Package ‘SimPhe’

October 12, 2022

Type  Package
Title  Tools to Simulate Phenotype(s) with Epistatic Interaction
Version  0.2.0
Author  Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz
Maintainer  Beibei Jiang <beibei_jiang@psych.mpg.de>
Description  Provides functions to simulate single or multiple, independent or correlated phenotype(s) with additive, dominance effects and their interactions. Also includes functions to generate phenotype(s) with specific heritability. Flexible and user-friendly options for simulation.

BugReports  https://github.com/beibeiJ/SimPhe/issues/new
URL  https://github.com/beibeiJ/SimPhe
License  GPL (>= 2)
LazyData  TRUE
RoxygenNote  6.1.0
VignetteBuilder  knitr
Suggests  testthat, knitr, rmarkdown
Encoding  UTF-8
NeedsCompilation  no
Repository  CRAN
Date/Publication  2018-09-13 14:40:02 UTC

R topics documented:

allele.freq ................................................................. 2
build.cor.phe .............................................................. 3
build.sd.matrix ............................................................. 4
calc.gene.var ............................................................... 5
calc.herit ................................................................. 6
check.snp.par ............................................................. 6
count.allele .............................................................. 7
epistasis.pars ............................................................ 8


Description

A dataset containing sample allele frequencies of SNPs. allele.freq.

Usage

data(allele.freq)

Format

A data frame with 6 rows and 3 columns(variables):

- **SNP** name of SNP
- **major.frequency** frequency of major allele
- **minor.frequency** frequency of minor allele

Frequencies of SNPs.
build.cor.phe  

Build correlated phenotypes

Description

Build correlated phenotypes

Usage

build.cor.phe(phen, corMtr, sdMtr = NULL, margin = 2, ...)

Arguments

- **phen**: a matrix or dataframe with the phenotypic information.
- **corMtr**: a correlation matrix.
- **sdMtr**: a matrix with the standard deviation, e.g., if the number of dimensions is 2, then it is \( \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \). If it is NULL (default), generate it based on the data from pheno.
- **margin**: a vector giving the subscript which the function will be applied over. E.g., for a matrix 1 indicates rows, 2 (default) indicates columns. Where pheno has named dimnames, it can be a character vector selecting dimension names.
- **...**: not used.

Value

a matrix with correlated phenotypes.

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de>

Examples

```r
x1 <- rnorm(4000, mean = 5, sd = 10)
x2 <- rnorm(4000, mean = 10, sd = 30)
x <- matrix(cbind(x1, x2), ncol = 2)

# test original correlation
cor.test(x[, 1], x[, 2])

# correlation matrix
corM <- matrix(c(1, 0.6, 0.6, 1), ncol = 2)

# standard deviation matrix
sdM <- matrix(c(10, 0, 0, 30), ncol = 2)

# build correlation
```
x.new <- build.cor.phe(x, corM, sdM)

# check mean and standard deviation of new data set
apply(x.new, 2, mean)
apply(x.new, 2, sd)

# test correlation
cor.test(x.new[, 1], x.new[, 2])

---

**build.sd.matrix**

*build a matrix with standard deviation*

**Description**
Build a matrix with standard deviation.

**Usage**

`build.sd.matrix(x, margin = 2, ...)`

**Arguments**

- `x` a matrix or dataframe.
- `margin` an integer specifying the dimension that standard deviation will be computed for. 1 indicates rows, 2 indicates columns. Default is 2.
- `...` not used

**Value**

a matrix with the standard deviation. If the number of dimensions is 2, then it is $\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$.

**Author(s)**

Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz <puetz@psych.mpg.de>

**Examples**

```r
x1 <- rnorm(4000, mean = 5, sd = 10)
x2 <- rnorm(4000, mean = 10, sd = 30)
x <- matrix(cbind(x1, x2), ncol = 2)
build.sd.matrix(x)
```
**calc.gene.var**

**Genetic Variance**

Description

Calculate the total genetic variance

Usage

```r
calc.gene.var(gene.coef, freq, Dskim = 0, ...)
```

Arguments

- `gene.coef` a list with the coefficients of genetic effects. Each element includes 8 genetic parameters (regression coefficient) $\beta_{Gw_t}$, t in (1, 2, …, 8)
- `freq` a dataframe with the allele frequencies.
- `Dskim` the coefficient of linkage disequilibrium. Default is 0 (no LD).
- `...` not used.

Details

The genetic variance is calculated based on the genetic parameters $\beta_{Gw_t}$, t in (1, 2, …, 8).

as described in the publications by Kao and Cockerham:

  [http://www.genetics.org/content/160/3/1243.short](http://www.genetics.org/content/160/3/1243.short)


Value

genetic variance.

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de>

Examples

```r
calc.gene.var(gene.coefficients, allele.freq)
```
calc.hertit  

Calculate heritability

Description

Calculate heritability

Usage

calc.hertit(gene.coef, freq, noise.var, Dskim = 0, ...)

Arguments

gene.coef

a vector of 8 genetic parameters. Each element includes 8 genetic parameters
(regression coefficient) $\beta_{GW_t}$ in $(1, 2, \cdots, 8)$.

dreq

a dataframe with the allele frequencies.

noise.var

variance of noise to generate the random noise.

Dskim

the coefficient of linkage disequilibrium. Default is 0 (no LD).

...

not used.

Value

heritability.

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de>

check.snp.par  

Check the number of the SNPs involved in epistasis and main effects

Description

Check the number of the SNPs set in the simulation parameters.

Usage

check.snp.par(genetic.pars, nphe, ...)

Arguments

genetic.pars

a data.frame or a matrix containing the parameter information for the main ef-
tect: additive and dominance.

nphe

number of phenotypes.

...

not used.
count.allele

Details
The number of SNPs involved in main effects should be the same as the number of SNPs involved in epistasis.

Value
NULL — will stop if test fails.

Author(s)
Beibei Jiang <beibei_jiang@psych.mpg.de>

Examples
check.snp.par(genepars, nphe = 2)

---

count.allele Count allele frequencies

Description
Count major and minor allele frequencies.

Usage
count.allele(x, ...)

Arguments
x a vector of single SNP information (minor allele count for genotype).
... not used.

Value
a vector with major and minor allele frequency.

Author(s)
Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz <puetz@psych.mpg.de>

Examples
maf <- 0.1
x <- sample(0:2, 1000, replace = TRUE, prob = c((1-maf)^2, 2*(1-maf)*maf, maf^2))
table(x)
count.allele(x)
**epistasis.pars** *Parameter settings of epistatic effects.*

### Description

A dataset containing the parameter settings for epistatic effects. epistasis.pars.

### Usage

```r
data(epistasis.pars)
```

### Format

A data frame with 6 rows and 3 columns (variables):

- **SNPA**: first SNP
- **SNPB**: second SNP
- **additive_additive**: coefficient for additive-additive interaction
- **additive_dominance**: coefficient for additive-dominance interaction
- **dominance_additive**: coefficient for dominance-additive interaction
- **dominance_dominance**: coefficient for dominance-dominance interaction

**gene.coefficients** *Coefficients of genetic effects.*

### Description

A dataset containing the regression coefficients of genetic effects. gene.coefficients.

### Usage

```r
data(gene.coefficients)
```

### Format

A list with 3 elements:

- **epi.par1**: a data frame with 1 row and 10 variables:
  - **SNPA**: first SNP
  - **SNPB**: second SNP
  - **additiveA**: coefficient for additive effect at locus A
  - **dominanceA**: coefficient for dominance effect at locus A
  - **additiveB**: coefficient for additive effect at locus B
  - **dominanceB**: coefficient for dominance effect at locus B
gene.effect

Description
Get genetic effect for each individual based on the genotype.

Usage

gene.effect(geno, gene.coef, model = c("epistasis"), ...)

Arguments

geno          a data.frame or a matrix containing the genotype information.
gene.coef     a list with the coefficients of genetic effects.
model         a string specifying the genetic model to use for the simulation. Default is "epistasis".
...           not used.
Value

a data.frame including genetic effects.

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz <puetz@psych.mpg.de>

Examples

# genotype file: rows are individuals and columns are SNPs
fgeno.path <- system.file("extdata", "10SNP.txt", package="SimPhe")

# get genotype
geno <- read.geno(fgeno.path, ftype = "snp.head")

# take a look at geno and gene.coefficients
geno
gene.coefficients

genepars

Parameter settings for simulation.

Description

A dataset containing the parameters for a simulation. genepars.

Usage

data(genepars)

Format

A list with 7 elements:

P1mean  A data frame with 1 row and 1 variable:

mean  $\beta_0$: coefficient parameter of "basic" genetic effects in $G_{ij} = \beta_0 + \sum_{t=1}^{8} \beta_{G_w} w_{ij}$.

P1main  A data frame with 6 rows and 3 variables:

SNP  SNP name
additive  coefficient of additive effect
dominance  coefficient of dominance effect

P1epistasis  A data frame with 3 rows and 6 variables:

SNPA  first SNP
SNPB  second SNP
additive_additive  coefficient for additive-additive interaction
additive_dominance  coefficient for additive-dominance interaction
dominance_additive  coefficient for dominance-additive interaction
dominance_dominance  coefficient for dominance-dominance interaction

P1 heritability  A data frame with 1 row and 1 variable:
heritability  expected heritability

P2 mean  A data frame with 1 row and 1 variable:
mean  mean of genetic effect

P2 main  A data frame with 6 rows and 3 variables:
SNP  SNP name
additive  coefficient of additive effect
dominance  coefficient of dominance effect

P2 epistasis  A data frame with 3 rows and 6 variables:
SNPA  first SNP
SNPB  second SNP
additive_additive  coefficient for additive-additive interaction
additive_dominance  coefficient for additive-dominance interaction
dominance_additive  coefficient for dominance-additive interaction
dominance_dominance  coefficient for dominance-dominance interaction

---

**genetic.scale**  
*Genetic scales of epistatic model*

**Description**

Genetic scales of epistatic model (Cockerham model) based on F2 population.

**Usage**

```r
genetic.scale(SNPA = 0, SNPB = 0)
```

**Arguments**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNPA</td>
<td>encoded alleles for first SNP.</td>
</tr>
<tr>
<td>SNPB</td>
<td>encoded alleles for second SNP.</td>
</tr>
</tbody>
</table>
Details

Calculate the genetic scale for a given allele combination of two SNPs. There are 9 genotypes in an F2 population, so we need 8 genetic parameters to give a complete description of the values for the 9 genotypes. Under the assumption of Hardy-Weinberg and linkage equilibrium, Cockerham (1954)’s orthogonal partition of genetic variance leads to the definition of the genotypic value \( G_{ij} \)

\[
G_{ij} = \beta_0 + \sum_{t=1}^{8} \beta_{G_{wt}} w_{tij}
\]

by eight orthogonal scales or contrasts \( w_t \)’s, \( t \) in \( 1, 2, \ldots, 8 \). Four are marginal scales and four are interaction scales. Marginal scales (defined by Model I for an F2 population) are called linear and quadratic scales (additive and dominance scales in genetic terms). Correspondingly, the interaction scales are

- \( w_1 \) additive for locus A;
- \( w_2 \) dominance for locus A;
- \( w_3 \) additive for locus B;
- \( w_4 \) dominance for locus B;
- \( w_5 = w_1 \times w_3 \) linear \( \times \) linear, additive \( \times \) additive;
- \( w_6 = w_1 \times w_4 \) linear \( \times \) quadratic, additive \( \times \) dominance;
- \( w_7 = w_2 \times w_3 \) quadratic \( \times \) linear, dominance \( \times \) additive;
- \( w_8 = w_2 \times w_4 \) quadratic \( \times \) quadratic, dominance \( \times \) dominance.

SNPs are encoded by (0, 1, 2):

- 0 means homozygous with major alleles;
- 1 means heterozygote;
- 2 means homozygous with minor alleles

E.g., the SNPs are encoded as 0: AA, 1: AG, and 2: GG, where 'A' represents the major allele and 'G' the minor allele.

Value

- a vector of genetic scales.

Author(s)

- Benno Pütz <puetz@mpipsykl.mpg.de> and Beibei Jiang <beibei_jiang@psych.mpg.de>

Examples

```
 geneti.c.scale(SNPA = 1, SNPB = 0)
```
get.freq

Description

Calculate the gene(allele) frequency for each of the SNPs.

Usage

```r
get.freq(geno, epi.pars, ...)
```

Arguments

- **geno**: a dataframe of genotype data: columns are the SNPs; lines are individuals.
- **epi.pars**: a data.frame or a matrix containing the parameter information for epistatic effect: additive × additive, additive × dominance, dominance × additive, and dominance × dominance.
- **...**: not used

Value

a dataframe with allele frequencies (major and minor).

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de>

Examples

```r
# genotype file: rows are individuals and columns are SNPs
fgeno.path <- system.file("extdata", "10SNP.txt", package="SimPhe")

# get genotype
geno <- read.geno(fgeno.path, ftype = "snp.head")

get.freq(geno, epistasis.pars)
```
get.gene.coef

Get the coefficients of genetic effect

Description
Get the coefficients of genetic effects.

Usage
get.gene.coef(main.pars, epi.pars, model = c("epistasis"), ...)

Arguments
main.pars a data.frame or a matrix containing the parameters for the main effect: additive and dominance.
epi.pars a data.frame or a matrix containing the parameters for the epistatic effect: additive \times additive, additive \times dominance, dominance \times additive, dominance \times dominance.
model a string show the genetic model to use for simulation. Default is "epistasis"
... not used.

Value
a list with the coefficients of genetic effects.

Author(s)
Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz <puetz@psych.mpg.de>

Examples
# take a look at the settings of coefficients for main effects
maineff.pars

# take a look at the settings of coefficients for interactive effects
epistasis.pars

# get a vector of gene coefficients
get.gene.coef(maineff.pars, epistasis.pars)
**get.noise.var**

*Description*

Give suggestion on the parameter setting of noise variance according to the expected heritability.

*Usage*

```r
get.noise.var(gene.coef, freq, heritability, Dskim = 0, ...)
```

*Arguments*

- `gene.coef`: a list including the coefficients of genetic effects. Each element includes 8 genetic parameters (regression coefficient) $\beta_{Gw_t}$, $t$ in (1, 2, ..., 8)
- `freq`: a dataframe with the allele frequencies.
- `heritability`: expected heritability.
- `Dskim`: the coefficient of linkage disequilibrium. Default is 0 (no LD).
- `...`: not used.

*Value*

variance of noise to generate the random noise.

*Author(s)*

Beibei Jiang <beibei_jiang@psych.mpg.de>

*Examples*

```r
get.noise.var(gene.coefficients, allele.freq, 0.5)
```

---

**list2frame**

*Description*

Convert list to data.frame.

*Usage*

```r
list2frame(x, ...)
```
Arguments

x  a list. In this package, it is used as a list includes simulated phenotypes. One element per block. Each element is a dataframe with SNP names and individuals.

Value

a data.frame.

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de>

Examples

```r
x <- list(test1=matrix(rnorm(1000), ncol=2), test2=matrix(rnorm(1000), ncol=2))
str(x)
x.new <- list2frame(x)
str(x)
```

<table>
<thead>
<tr>
<th>maineff.pars</th>
<th>Parameter settings of main effects (additive effect and dominance).</th>
</tr>
</thead>
</table>

Description

A dataset containing the parameter settings for main effects. maineff.pars.

Usage

data(maineff.pars)

Format

A data frame with 6 rows and 3 columns(variables):

- **SNP**  name of SNP
- **additive**  coefficient of additive effect
- **dominance**  coefficient of dominance effect
**pars.writer**

*Write current parameters to file*

---

**Description**

Write out the information about the parameters for simulation.

**Usage**

```r
pars.writer(genetic.pars, fname = "usedpars.txt", ...)```

**Arguments**

- `genetic.pars`: list of simulation parameters. One element per block of file. Each element is a dataframe with SNP names and model parameters.
- `fname`: filename of the setting for simulation (for recording).
- `...`: not used.

**Author(s)**

Beibei Jiang <beibei_jiang@psych.mpg.de>

**Examples**

```r
pars.writer(genepars)
```

---

**phe.writer**

*Write phenotypes*

---

**Description**

Write out the simulated phenotypes.

**Usage**

```r
phe.writer(phe, onefile = TRUE, fname = "simu.pheno", ...)```

**Arguments**

- `phe`: a data.frame or a matrix of simulated phenotypes. Each column is a phenotype.
- `onefile`: whether to create just one file for all phenotypes (default) or one per phenotype
- `fname`: filename of the phenotype(s).
- `...`: not used.
Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz <puetz@psych.mpg.de>

Examples

```r
phe <- matrix(rnorm(1000), ncol = 2)
colnames(phe) <- c("p1", "p2")
phe.writer(phe)
```

read.geno

---

**Read genotype data based on the file format**

Description

Read genotype data.

Usage

```r
read.geno(fname = NULL, verbose = getOption("verbose"), run = TRUE,
cleanup = TRUE, ftype = c("ind.head", "plink", "snp.head"),
plink.path = NULL, ...)
```

Arguments

- `fname` a string specifying the file to read genotype information from
- `verbose` when set show the commands that are to be called through system.
- `run` when set (default) execute the system calls.
- `cleanup` when set (default) remove intermediate files before returning.
- `ftype` genotype file format, it accepts three options:
  - "plink": plink format (.bed, .bim, .fam or .map, .ped);
  - "ind.head": columns are the individuals and lines are SNPs;
  - "snp.head": columns are SNPs and lines are individuals.

For "plink", `fgeno` needs to be given without suffix and `plink.path` may need to be assigned by the user because plink will be run from within SimPhe. More detail see plink.path. For the other options, `fgeno` should be the full name (with suffix and path if necessary) of the genotype file. Of course, this does not apply if `fgeno` is provided as a data frame.

- `plink.path` path of plink executable. Only needed when the `ftype` is "plink". Default is NULL. The function will detect the plink path with `system("where plink")` for Windows users and `system("which plink")` for Linux and MacOS users. But there is no guarantee that the commands work on all devices. If the path cannot be determined or the executable cannot be called from `read.geno`, then users have to try other formats.

... not used.
Details

If it is plink file format (.bed, .bim, .fam), make sure that plink has already been installed in the system.

Value

A dataframe of genotype data: columns are the SNPs; rows are individuals.

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz <puetz@psych.mpg.de>

Examples

### "snp.head" genotype file: rows are individuals and columns are SNPs
# get full path of example file
fgeno.path <- system.file("extdata", "10SNP.txt", package="SimPhe")

geno <- read.geno(fgeno.path, ftype = "snp.head")
head(geno)

### "plink" genotype file: 1).map and .ped; 2).bed, .fam and .bim
# get directory of plink example file
fpath <- strsplit(fgeno.path, "10SNP.txt")

#### Note: before run this example, specify your installation path of plink ####
# geno <- read.geno(paste0(fpath, "bdemo"), ftype = "plink", plink.path = "user's plink path")

---

read.simu.pars  Read parameters

Description

Read file specifying the simulation parameters.

Usage

read.simu.pars(file = NULL, ...)

Arguments

file the file of parameters settings
...

not used
Details

File format: please follow the example of the simupars.txt file found in the inst/extdata/ directory of the package (run system.file("extdata", "simupars.txt", package="SimPhe") to see the full path), blank lines are ignored. The file consists of three or four blocks for each phenotype: mean, main, epistasis, and (optionally) heritability. Each block is started by a line of the form `[blockname]` followed by the parameters for the block, e.g., for the first phenotype,

```
[P1mean ]
mean β0: coefficient parameter of "basic" genetic effects in G_{ij} = β_0 + ∑_{t=1}^8 β_{G wt} w_{tij}.

[P1main ]
SNP SNP name
additive coefficient of additive effect
dominance coefficient of dominance effect

[P1epistasis ]
SNPA first SNP name
SNPB second SNP name
additive_additive coefficient for additive-additive interaction
additive_dominance coefficient for additive-dominance interaction
dominance_additive coefficient for dominance-additive interaction
dominance_dominance coefficient for dominance-dominance interaction

[P1heritability ]
heritability expected heritability
```

For each block the expected columns and the respective meanings are given. Similar blocks need to be provided for the other phenotype(s): 

```
[P2mean ]
[P2main ]
[P2epistasis ]
```

and so on.

Value

a list of simulation parameters. One element per block of file. Each element is a dataframe with SNP names and model parameters

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz <puetz@psych.mpg.de>

Examples

```r
# simulation parameters:
FPAR.path <- system.file("extdata", "simupars.txt", package="SimPhe")

# pars <- read.simu.pars(FPAR.path)
```
**Description**

Extract (sub)strings matching regex pattern.

**Usage**

`regextract(x, pattern, ...)`

**Arguments**

- `x`: a character vector.
- `pattern`: regular expression to be found in `x`.
- `...`: not used.

**Details**

Extract the substrings of `x` that match the regex pattern `pattern`. The pattern may contain groups (enclosed in parentheses) which will result in further substrings extracted.

Derived from the help on `regmatches`, take a look at the help there.

**Value**

A character (string) matrix where the first column contains the global match for the pattern, each pair of `()` will result in another column with the respective match.

**Author(s)**

Benno Pütz <puetz@psych.mpg.de>

**Examples**

```r
s <- "Test: A1 BC23 DEF456"
pattern = "([[:alpha:]]+)([[:digit:]]+)"
regextract(s, pattern)

# equivalent to this example from the help page for grep()
lapply(regmatches(s, gregexpr(pattern, s)), function(e) regmatches(e, regexec(pattern, e)))
```
**sim.phe**

*Simulation for phenotypes (SimPhe main process)*

### Description

Main process of simulation for phenotypes

### Usage

```r
sim.phe(sim.pars = NULL, fgeno = NULL, ftype = c("ind.head", "plink", "snp.head"), fwrite = TRUE, fphename = "simu.pheno", fusepar = "usedpars.txt", seed = NA, Dskim = 0, noise.var = 1, pattern = "[[:alpha:]]+", plink.path = system("which plink"), genetic.model = "epistasis", ...)
```

### Arguments

- **sim.pars**
  
a prepared list containing the parameters settings for simulation or a file of parameters settings. Please set your own parameters following the same structure as the object genepars or as the file simpars.txt (you could find example file system.file("extdata","simupars.txt",package="SimPhe")).

To specify heritability, there are two ways: one is to set heritability in parameter file or in the prepared list object which will further pass to sim.pars. Another way is to set noise.var by using function `get.noise.var` given specify heritability.

List format: please follow the example of the object genepars. The meaning of each element in the list is similar like the file format description below.

File format: please follow the example of the simpars.txt file found in the inst/extdata/directory of the package (run `system.file("extdata","simupars.txt",package="SimPhe")` to get the path to the file), blank lines are ignored. The file consists of three or four blocks for each phenotype (the number of blocks depends on user): mean, main and epistasis, sometimes heritability. Each block is started by a line of the form `'[blockname]'` followed by the parameter setting for the block, e.g. for first phenotype,

```
[P1mean ]
  mean \( \beta_0 \): coefficient parameter of "basic" genetic effects in \( G_{ij} = \beta_0 + \sum_{t=1}^{8} \beta_{G_{wt}} w_{ij} \).
```

```
[P1main ]
  SNP  SNP name
  additive  coefficient of additive effect
```
SimPhe

- **dominance** coefficient of dominance effect
- **[P1epistasis]**
  - **SNPA** first SNP
  - **SNPB** second SNP
- **additive_additive** coefficient for additive-additive interaction
- **additive_dominance** coefficient for additive-dominance interaction
- **dominance_additive** coefficient for dominance-additive interaction
- **dominance_dominance** coefficient for dominance-dominance interaction
- **P1heritability** expected heritability

Similar meanings for "[P2mean]", "[P2main]", "[P2epistasis]", and so on.

- **fgeno** file to read genotype information from or pre-read data.frame with that information (matching the output format of `read.geno`).
- **ftype** genotype file format, it accepts three options:
  - "plink": plink format (.bed, .bim, .fam or .map, .ped);
  - "ind.head": columns are the individuals and lines are SNPs;
  - "snp.head": columns are SNPs and lines are individuals.

For "plink", `fgeno` needs to be given without suffix and `plink.path` may need to be assigned by the user because `plink` will be run from within SimPhe. More detail see `plink.path`. For the other options, `fgeno` should be the full name (with suffix and path if necessary) of the genotype file. Of course, this does not apply if `fgeno` is provided as a data frame.

- **fwrite** logical. Write out file (simulated data) or not. If TRUE (default), simulated phenotypes will be written, respectively.
- **fphename** filename of the phenotype(s). Default is "simu.pheno".
- **fusepar** filename of the setting for simulation (for recording). Default is "usedpars.txt".
- **seed** an integer used for set.seed(). Default is NA.
- **Dskim** the coefficient of linkage disequilibrium. Default is 0 (no LD).
- **noise.var** variance for random noise. Default is 1. Note that this is overridden by the heritability setting in the simulation parameter file. If heritability is given in parameter file then `noise.var` will not work.
- **pattern** ignore pattern for detecting the phenotype index from the parameter names. Default is "[[[:alpha:]]]+" which means letters.
- **plink.path** path of plink executable. Only needed when the `ftype` is "plink". Default is NULL. The function will detect the plink path with `system("where plink")` for Windows users and `system("which plink")` for Linux and MacOS users. But there is no guarantee that the commands work on all devices. If the path cannot be determined or the executable cannot be called from `read.geno`, then users have to try other formats.
- **genetic.model** a string show the genetic model to use for simulation. Default is "epistasis".
- ... not used.
Details

further discussion on pattern

Value

a data.frame with the simulated phenotype(s) where the column(s) refer to different phenotype(s) and rows to individuals.

Author(s)

Beibei Jiang <beibei_jiang@psych.mpg.de> and Benno Pütz <puetz@psych.mpg.de>

Examples

```
#### file path of example
# simulation parameters:
fpars.path <- system.file("extdata", "simupars.txt", package="SimPhe")

# genotype file: rows are individuals and columns are SNPs
fgeno.path <- system.file("extdata", "10SNP.txt", package="SimPhe")

#### instead of a parameter file, prepared list like genepars also works

# prepare genepars

#### simulate phenotype(s)
phe <- sim.phe(sim.pars = fpar.path, fgeno = fgeno.path, ftype = "snp.head", fwrite = FALSE)
# or
phe <- sim.phe(sim.pars = genepars, fgeno = fgeno.path, ftype = "snp.head", fwrite = FALSE)

# the simulated phenotype(s)
str(phe)
head(phe)
```

---

**specify.pars**

*Get the parameters of main/epistatic effects per phenotype*

Description

Get the parameters of main/epistatic effects per phenotype.

Usage

```
specify.pars(genetic.pars, effect.type = c("main", "epistasis"),
              phe.index = 1, ...)
```
**specify.pars**

**Arguments**

- `genetic.pars` a data.frame or a matrix containing the parameters information for main effect: additive and dominance.
- `effect.type` a string naming the type of the genetic effects (accepts either "main" or "epistasis").
- `phe.index` a integer indicating the phenotype. Default is 1.
- `...` not used.

**Value**

a data.frame or a matrix containing the parameters information for epistatic effect: additive × additive, additive × dominance, dominance × additive, dominance × dominance.

**Author(s)**

Beibei Jiang <beibe_jiang@psych.mpg.de>

**Examples**

```r
# get parameters of coefficients for main effects
specify.pars(genepars, effect.type = "main")
```

```r
# get parameters of coefficients for interactive effects
specify.pars(genepars, effect.type = "epistasis")
```
Index

* datasets
  allele.freq, 2
  epistasis.pars, 8
  gene.coefficients, 8
  genepars, 10
  maineff.pars, 16

allele.freq, 2
build.cor.phe, 3
build.sd.matrix, 4
calc.gene.var, 5
calc.herit, 6
check.snp.par, 6
count.allele, 7
epistasis.pars, 8
gene.coefficients, 8
gene.effect, 9
genepars, 10
genetic.scale, 11
get.freq, 13
get.gene.coef, 14
get.noise.var, 15, 22
list2frame, 15
maineff.pars, 16
pars.writer, 17
phe.writer, 17
read.geno, 18, 23
read.simu.pars, 19
regextract, 21
regmatches, 21
sim.phe, 22
specify.pars, 24