

# Package: ALLSPICER (via r-universe)

October 18, 2024

**Type** Package

**Title** ALLelic Spectrum of Pleiotropy Informed Correlated Effects

**Version** 0.1.9

**Maintainer** Wenhan Lu <wlu@broadinstitute.org>

**Description** Provides statistical tools to analyze heterogeneous effects of rare variants within genes that are associated with multiple traits. The package implements methods for assessing pleiotropic effects and identifying allelic heterogeneity, which can be useful in large-scale genetic studies. Methods include likelihood-based statistical tests to assess these effects. For more details, see Lu et al. (2024) <doi:10.1101/2024.10.01.614806>.

**License** MIT + file LICENSE

**Encoding** UTF-8

**Imports** dplyr, magrittr, readr, mvtnorm, stats

**RoxygenNote** 7.3.2

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**Config/testthat/edition** 3

**NeedsCompilation** no

**Author** Wenhan Lu [aut, cre]

**Date/Publication** 2024-10-16 17:50:06 UTC

**Repository** https://wlu04.r-universe.dev

**RemoteUrl** https://github.com/cran/ALLSPICER

**RemoteRef** HEAD

**RemoteSha** 015023f015c64134e56b7d8ed51dc93243793baa

## Contents

ALLSPICE	2
ALLSPICE_simulation	3

format_ALLSPICE_data . . . . .	4
get_ac_mat . . . . .	4
get_af_mat . . . . .	5
get_beta_hat . . . . .	6
get_c_hat . . . . .	6
get_geno_mat . . . . .	7
get_likelihood_test_stats . . . . .	8
get_mle_beta . . . . .	9
get_pheno_pair . . . . .	10
get_single_geno . . . . .	10
get_true_beta . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

ALLSPICE

*ALLSPICE*

---

## Description

ALLSPICE (ALLelic Spectrum of Pleiotropy Informed Correlated Effects)

## Usage

```
ALLSPICE(
  data,
  pheno_corr,
  n_ind,
  gene = "GENENAME",
  pheno1 = "PHENO1",
  pheno2 = "PHENO2",
  beta1_field = "BETA1",
  beta2_field = "BETA2",
  af_field = "AF"
)
```

## Arguments

data	Input data with number of rows indicating number of variants, three columns are required: 1) effect sizes of variants for phenotype 1, 2) effect sizes of variants for phenotype 2, 3) allele frequency of variants Note: this should include variants from ONE gene that is associated with the two phenotypes, preferably of the SAME functional category after being filtered to variants with allele frequency below a certain threshold (e.g. 1e-4)
pheno_corr	phenotypic correlation between the two phenotypes being tested
n_ind	total number of individuals
gene	name of the gene being tested, default 'GENENAME'
pheno1	descriptive name of phenotype 1, default 'PHENO1'

pheno2	descriptive name of phenotype 2, default 'PHENO2'
beta1_field	field name for effect sizes of variants on phenotype 1, default 'BETA1'
beta2_field	field name for effect sizes of variants on phenotype 2, default 'BETA2'
af_field	field name for allele frequencies of variants, default 'AF'

**Value**

A list of summary statistics from ALLSPICE test including phenotype names, gene names, MLE of slope  $c$ , ALLSPICE test statistic -  $\lambda$ , pvalue from a chi-square distribution, total number of variants being tested

**Examples**

```
data <- data.frame(x = rnorm(10), y = rnorm(10), z = runif(10, 0,1))
ALLSPICE(data,pheno_corr=0.5,n_ind=10000,beta1_field='x',beta2_field='y',af_field='z')
```

---

ALLSPICE\_simulation    *ALLSPICE\_simulation*

---

**Description**

Simulate data and run ALLSPICE

**Usage**

```
ALLSPICE_simulation(n_ind, n_var, c, r, pi, sigma, mle = TRUE, null = TRUE)
```

**Arguments**

n_ind	total number of individuals
n_var	total number of variants
c	slope between the two sets of variant effect sizes, only applicable when 'null' == TRUE
r	phenotypic correlation between the two phenotypes
pi	probability of variant of having no effect on the phenotype
sigma	variance of the two sets of effect sizes
mle	whether to use MLE of $c$ to compute the test statistic, use true $c$ value if FALSE
null	whether to simulate data under the null hypothesis (no linear relationship) or the alternative hypothesis

**Value**

A list of two pieces of results: 1) ALLSPICE test results 2) effect size table: true effect size simulated, effect size estimate from linear model, effect size estimated from MLE

**Examples**

```
ALLSPICE_simulation(n_ind=10000, n_var=100, c=0.6, r=0.5, pi=0.5, sigma=1, mle = TRUE, null=TRUE)
```

---

```
format_ALLSPICE_data  format_ALLSPICE_data
```

---

**Description**

data formatting function: format raw data to be loaded into ALLSPICE

**Usage**

```
format_ALLSPICE_data(data, beta1_field, beta2_field, af_field)
```

**Arguments**

data	raw input data
beta1_field	field name of effect size for the first phenotype
beta2_field	field name of effect size for the second phenotype
af_field	field name of allele frequency information

**Value**

a data frame containing effect sizes of variants on two phenotypes and their allele frequency information

**Examples**

```
data <- data.frame(x = rnorm(10), y = rnorm(10), z = runif(10, 0,1))
data <- format_ALLSPICE_data(data=data, beta1_field = 'x', beta2_field = 'y', af_field = 'z')
```

---

```
get_ac_mat  get_ac_mat
```

---

**Description**

simulation function: simulate allele count information for 'n\_var' variants, with a maximum allele count 'max\_cnt'

**Usage**

```
get_ac_mat(n_var, max_cnt = 100)
```

**Arguments**

`n_var` total number of variants  
`max_cnt` maximum allele count, default 100

**Value**

A '`n_var`'x'`n_var`' diagonal matrix of allele count information for '`n_var`' variants

**Examples**

```
ac_mat <- get_ac_mat(n_var=100, max_cnt = 100)
```

---

<code>get_af_mat</code>	<i>get_af_mat</i>
-------------------------	-------------------

---

**Description**

simulation function: compute allele frequency information variants with allele counts stored in diagonal matrix 'AC' from a population of sample size '`n_ind`'

**Usage**

```
get_af_mat(AC, n_ind)
```

**Arguments**

`AC` a diagonal matrix of allele count information for all variants  
`n_ind` total number of individuals in the population

**Value**

A '`n_var`'x'`n_var`' diagonal matrix of allele frequency information for '`n_var`' (dimension of 'AC') variants

**Examples**

```
af_mat <- get_af_mat(AC = c(20, 50, 10, 1, 5), n_ind = 10000)
```

---

get\_beta\_hat                    *get\_beta\_hat*

---

### Description

simulation function: compute effect sizes estimated form linear regression model

### Usage

```
get_beta_hat(Y, X, A, n_ind)
```

### Arguments

Y	phenotype information
X	genotype information
A	Allele frequency information
n_ind	total number of individuals

### Value

A 2x'n\_var' matrix of estimated effect size information (first row corresponds to the first phenotype, second row corresponds to the second phenotype)

### Examples

```
AC <- get_ac_mat(n_var=100)
A <- get_af_mat(AC=AC, n_ind=10000)
X <- get_geno_mat(AC, n_ind=10000)
b <- get_true_beta(n_var=100, c=0.6, pi=0.5, sigma=1, null=TRUE)
Y <- get_pheno_pair(b=b, X=X, r=0.5)
b_hat <- get_beta_hat(Y=Y, X=X, A=A, n_ind=10000)
```

---

get\_c\_hat                    *get\_c\_hat*

---

### Description

ALLSPICE function: compute the slope 'c' that maximize the likelihood (maximum likelihood estimate - MLE)

### Usage

```
get_c_hat(b1_hat, b2_hat, A, r)
```

**Arguments**

b1_hat	estimated effect size of the first phenotype across all variants
b2_hat	estimated effect size of the second phenotype across all variants
A	Allele frequency information
r	phenotypic correlation between the two phenotypes

**Value**

the MLE of slope between two sets of effect sizes

**Examples**

```
AC <- get_ac_mat(n_var=100)
A <- get_af_mat(AC=AC, n_ind=10000)
X <- get_geno_mat(AC, n_ind=10000)
b <- get_true_beta(n_var=100, c=0.6, pi=0.5, sigma=1, null=TRUE)
Y <- get_pheno_pair(b=b, X=X, r=0.5)
b_hat <- get_beta_hat(Y=Y, X=X, A=A, n_ind=10000)
b1_hat <- matrix(b_hat[1, ], nrow = 1)
b2_hat <- matrix(b_hat[2, ], nrow = 1)
c_hat <- get_c_hat(b1_hat=b1_hat, b2_hat=b2_hat, A=A, r=0.5)
```

---

get\_geno\_mat

*get\_geno\_mat*

---

**Description**

simulation function: simulate genotype information for a set of loci with allele counts 'AC'

**Usage**

```
get_geno_mat(AC, n_ind)
```

**Arguments**

AC	allele counts of loci (length 'm')
n_ind	total number of indications

**Value**

An 'n\_ind'x'm' matrix of genotype information of 'n\_ind' individuals and 'm' variants

**Examples**

```
geno_mat <- get_geno_mat(AC = c(20, 50, 10, 1, 5), n_ind = 10000)
```

---

```
get_likelihood_test_stats
      get_likelihood_test_stats
```

---

**Description**

ALLSPICE function: compute the maximum likelihood ratio of the ALLSPICE test statistic

**Usage**

```
get_likelihood_test_stats(n_ind, r, b1_hat, b2_hat, c, A)
```

**Arguments**

n_ind	total number of individuals
r	phenotypic correlation between the two phenotypes
b1_hat	estimated effect size of the first phenotype across all variants
b2_hat	estimated effect size of the second phenotype across all variants
c	MLE of the slope between the two sets of variant effect sizes
A	Allele frequency information

**Value**

A single numeric value representing the test statistic of ALLSPICE (maximum likelihood ratio)

**Examples**

```
AC <- get_ac_mat(n_var=100)
A <- get_af_mat(AC=AC, n_ind=10000)
X <- get_genomat(AC, n_ind=10000)
b <- get_true_beta(n_var=100, c=0.6, pi=0.5, sigma=1, null=TRUE)
Y <- get_pheno_pair(b=b, X=X, r=0.5)
b_hat <- get_beta_hat(Y=Y, X=X, A=A, n_ind=10000)
b1_hat <- matrix(b_hat[1, ], nrow = 1)
b2_hat <- matrix(b_hat[2, ], nrow = 1)
c_hat <- get_c_hat(b1_hat=b1_hat, b2_hat=b2_hat, A=A, r=0.5)
lambda <- get_likelihood_test_stats(n_ind=10000, r=0.5, b1_hat=b1_hat, b2_hat=b2_hat, c=c_hat, A=A)
```



---

get_mle_beta	<i>get_mle_beta</i>
--------------	---------------------

---

### Description

ALLSPICE function: compute the effect size estimates that maximize the likelihood (maximum likelihood estimate - MLE) conditioning on c

### Usage

```
get_mle_beta(b1_hat, b2_hat, c, r, null = TRUE)
```

### Arguments

b1_hat	estimated effect size of the first phenotype across all variants
b2_hat	estimated effect size of the second phenotype across all variants
c	slope between the two sets of variant effect sizes, only applicable when 'null' == TRUE
r	phenotypic correlation between the two phenotypes
null	whether to simulate data under the null hypothesis (no linear relationship) or the alternative hypothesis

### Value

A 2x'n\_var' matrix of MLE estimated effect size information (first row corresponds to the first phenotype, second row corresponds to the second phenotype)

### Examples

```
AC <- get_ac_mat(n_var=100)
A <- get_af_mat(AC=AC, n_ind=10000)
X <- get_geno_mat(AC, n_ind=10000)
b <- get_true_beta(n_var=100, c=0.6, pi=0.5, sigma=1, null=TRUE)
Y <- get_pheno_pair(b=b, X=X, r=0.5)
b_hat <- get_beta_hat(Y=Y, X=X, A=A, n_ind=10000)
b1_hat <- matrix(b_hat[1, ], nrow = 1)
b2_hat <- matrix(b_hat[2, ], nrow = 1)
b_mle <- get_mle_beta(b1_hat=b1_hat, b2_hat=b2_hat, c=0.6, r=0.5, null=TRUE)
```

---

get\_pheno\_pair      *get\_pheno\_pair*

---

**Description**

simulation function: simulate true phenotype values of a pair of phenotypes

**Usage**

```
get_pheno_pair(b, X, r)
```

**Arguments**

b	true effect size matrix of variants on the two phenotypes
X	genotype matrix
r	phenotypic correlation between the two phenotypes

**Value**

A 2x'n\_ind' matrix of phenotype information (first row corresponds to the first phenotype, second row corresponds to the second phenotype)

**Examples**

```
AC <- get_ac_mat(n_var=100)
X <- get_geno_mat(AC, n_ind=10000)
b <- get_true_beta(n_var=100, c=0.6, pi=0.5, sigma=1, null=TRUE)
Y <- get_pheno_pair(b=b, X=X, r=0.5)
```

---

get\_single\_geno      *get\_single\_geno*

---

**Description**

simulation function: simulate genotype information for one locus, where 'cnt' samples out of 'n\_ind' has the mutation

**Usage**

```
get_single_geno(cnt, n_ind)
```

**Arguments**

cnt	number of individuals with the mutation
n_ind	total number of individuals

**Value**

A binary vector representing the genotype information of 'n\_ind' individuals for a particular locus, where 'cnt' entries has value 1.

**Examples**

```
geno <- get_single_geno(cnt = 100, n_ind = 10000)
```

---

get_true_beta	<i>get_true_beta</i>
---------------	----------------------

---

**Description**

simulation function: simulate true effect size information of 'n\_var' variants for two phenotypes

**Usage**

```
get_true_beta(n_var, c, pi, sigma, null = TRUE)
```

**Arguments**

n_var	total number of variants
c	slope between the two sets of variant effect sizes, only applicable when 'null' == TRUE
pi	probability of variant of having no effect on the phenotype
sigma	variance of the two sets of effect sizes
null	whether to simulate data under the null hypothesis (no linear relationship) or the alternative hypothesis

**Value**

A 2x'n\_var' matrix of effect size information for 'n\_var' variants (first row corresponds to the first phenotype, second row corresponds to the second phenotype)

**Examples**

```
true_beta <- get_true_beta(n_var=100, c=0.6, pi=0.5, sigma=1, null=TRUE)
```

# Index

ALLSPICE, [2](#)

ALLSPICE\_simulation, [3](#)

format\_ALLSPICE\_data, [4](#)

get\_ac\_mat, [4](#)

get\_af\_mat, [5](#)

get\_beta\_hat, [6](#)

get\_c\_hat, [6](#)

get\_geno\_mat, [7](#)

get\_likelihood\_test\_stats, [8](#)

get\_mle\_beta, [9](#)

get\_pheno\_pair, [10](#)

get\_single\_geno, [10](#)

get\_true\_beta, [11](#)