

# Package ‘SEAGLE’

November 5, 2021

**Type** Package

**Title** Scalable Exact Algorithm for Large-Scale Set-Based  
Gene-Environment Interaction Tests

**Version** 1.0.1

**Description** The explosion of biobank data offers immediate opportunities for gene-environment (GxE) interaction studies of complex diseases because of the large sample sizes and rich collection in genetic and non-genetic information. However, the extremely large sample size also introduces new computational challenges in GxE assessment, especially for set-based GxE variance component (VC) tests, a widely used strategy to boost overall GxE signals and to evaluate the joint GxE effect of multiple variants from a biologically meaningful unit (e.g., gene).

We present 'SEAGLE', a Scalable Exact Algorithm for Large-scale Set-based GxE tests, to permit GxE VC test scalable to biobank data. 'SEAGLE' employs modern matrix computations to achieve the same “exact” results as the original GxE VC tests, and does not impose additional assumptions nor relies on approximations. 'SEAGLE' can easily accommodate sample sizes in the order of  $10^5$ , is implementable on standard laptops, and does not require specialized equipment.

The accompanying manuscript for this package can be found at Chi, Ipsen, Hsiao, Lin, Wang, Lee, Lu, and Tzeng. (2021+) <[arXiv:2105.03228](https://arxiv.org/abs/2105.03228)>.

**URL** <https://github.com/jocelynchi/SEAGLE>

**License** GPL-3

**Depends** R (>= 3.5.0), Matrix, CompQuadForm

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Suggests** rmarkdown, knitr

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Jocelyn Chi [aut, cre],  
Ilse Ipsen [aut],  
Jung-Ying Tzeng [aut]

**Maintainer** Jocelyn Chi <jocetchi@gmail.com>

**Repository** CRAN

**Date/Publication** 2021-11-05 21:40:02 UTC

## R topics documented:

applyAt . . . . .	2
cosihap . . . . .	3
estimate.vc . . . . .	3
makeSimData . . . . .	4
prep.SEAGLE . . . . .	5
Rinv.AtG . . . . .	6
Rinv.u . . . . .	6
SEAGLE . . . . .	7
Vinv . . . . .	8
<b>Index</b>	<b>9</b>

---

applyAt	<i>Function for applying <math>t(A)</math> on the left for REML EM</i>
---------	--

---

### Description

Function for applying  $t(A)$  on the left for REML EM

### Usage

```
applyAt(qrXtilde, RHS)
```

### Arguments

qrXtilde	Object from QR decomposition of Xtilde
RHS	Object on right hand side of null of Xtilde <sup>T</sup>

### Value

Matrix or vector resulting from left multiplication of  $A$  with matrix or vector input RHS

---

`cosihap`*Synthetic haplotype data generated from COSI software*

---

**Description**

A dataset containing 10,000 haplotypes of SNP sequences mimicking the European population generated from the COSI software

**Usage**

```
data(cosihap)
```

**Format**

An object of class `dgCMatrix` with 10000 rows and 604 columns.

**Source**

<https://genome.cshlp.org/content/15/11/1576.abstract>

---

`estimate.vc`*REML EM Algorithm*

---

**Description**

REML EM algorithm for estimating variance components

**Usage**

```
estimate.vc(  
  y,  
  Xtilde,  
  qrXtilde,  
  beta,  
  G,  
  init.sigma = 0.5,  
  init.tau = 0.5,  
  tol = 0.001,  
  maxiters = 1000  
)
```

**Arguments**

<code>y</code>	Vector of observed phenotypes
<code>Xtilde</code>	Matrix of covariates (first column contains the intercept, last column contains the E factor for studying the GxE effect)
<code>qrXtilde</code>	Object containing QR decomposition of Xtilde
<code>beta</code>	Coefficient vector for covariate matrix Xtilde
<code>G</code>	Matrix of genotype markers
<code>init.sigma</code>	Initial sigma input (Default is 0.5)
<code>init.tau</code>	Initial tau input (Default is 0.5)
<code>tol</code>	Tolerance for convergence (Default is 1e-3)
<code>maxiters</code>	Maximum number of iterations (Default is 1000)

**Value**

Estimates for tau and sigma

---

<code>makeSimData</code>	<i>Generate synthetic data according to a fixed effects model</i>
--------------------------	---

---

**Description**

This function generates synthetic from the fixed effects model described in the experimental studies portion of the paper.

**Usage**

```
makeSimData(  
  H,  
  n,  
  L = 100,  
  maf = 0.01,  
  gamma0 = 1,  
  gammaX = 1,  
  gammaE = 1,  
  gammaG,  
  gammaGE,  
  causal = 40,  
  seed = 12345  
)
```

**Arguments**

H	Matrix of haplotype data (e.g. cosihap)
n	Number of individuals
L	Number of SNPs in the G matrix (Default is 100), should be a value between 1 and 604
maf	Minor allele frequency (Default is 0.01)
gamma0	gamma0 Fixed effect coefficient for intercept (Default is 1)
gammaX	gammaX Fixed effect coefficient for confounding covariates (Default is 1)
gammaE	gammaE Fixed effect coefficient for E effect (Default is 1)
gammaG	gammaG Fixed effect coefficient for G main effect
gammaGE	gammaGE Fixed effect coefficient for GxE interaction effect
causal	Number of causal SNPs (default is 40)
seed	Seed (Default is 12345)

**Value**

Synthetic dataset containing y, X, E, G, epsilon, and number of causal SNPs

**Examples**

```
dat <- makeSimData(H=cosihap, n=500, L=10, gammaG=1, gammaGE=0, causal=4, seed=1)
```

---

```
prep.SEAGLE
```

*Prepare data for input into SEAGLE function*

---

**Description**

This function checks and formats data for input into SEAGLE function

**Usage**

```
prep.SEAGLE(y, X, intercept, E, G)
```

**Arguments**

y	Vector of observed phenotypes
X	Matrix of covariates without genetic marker interactions
intercept	1 if the first column of X is the all ones vector, 0 otherwise
E	E Vector of environmental covariates
G	G Matrix of genotype data

**Value**

List object containing prepared data for input into SEAGLE function

**Examples**

```
dat <- makeSimData(H=cosihap, n=500, L=10, gammaG=1, gammaGE=0, causal=4, seed=1)
objSEAGLE <- prep.SEAGLE(y=dat$y, X=dat$X, intercept=1, E=dat$E, G=dat$G)
```

---

Rinv.AtG

*Function for applying R inverse to AtG in REML EM algorithm*


---

**Description**

Function for applying R inverse to AtG in REML EM algorithm

**Usage**

```
Rinv.AtG(G, AtG, GtAAtG, tau, sigma)
```

**Arguments**

G	Matrix of genotype markers (size n x L)
AtG	AtG from pre-computation
GtAAtG	GtAAtG from pre-computation
tau	Variance component from G main effect
sigma	Variance component from model noise epsilon

**Value**

Matrix resulting from left multiplication of Rinv with input matrix AtG

---

Rinv.u

*Function for applying R inverse to u in REML EM algorithm*


---

**Description**

Function for applying R inverse to u in REML EM algorithm

**Usage**

```
Rinv.u(G, AtG, GtAAtG, GtAu, u, tau, sigma)
```

**Arguments**

G	Matrix of genotype markers (size n x L)
AtG	AtG from precomputation
GtAAAtG	GtAAAtG from precomputation
GtAu	GtAu from precomputation
u	u=Aty from REML EM
tau	Variance component from G main effect
sigma	Variance component from model noise epsilon

**Value**

Vector resulting from left multiplication of Rinv with input vector u

---

SEAGLE	<i>Compute score-like test statistic and p-value for GxE test with SEAGLE algorithm</i>
--------	---

---

**Description**

This function computes the score test statistic and corresponding p-value for the GxE test with the SEAGLE algorithm with input data that have been prepared with the prep.SEAGLE function

**Usage**

```
SEAGLE(obj.SEAGLE, init.tau = 0.5, init.sigma = 0.5, pv = "liu")
```

**Arguments**

obj.SEAGLE	Input data prepared with prep.SEAGLE function
init.tau	Initial estimate for tau (Default is 0.5)
init.sigma	Initial estimate for sigma (Default is 0.5)
pv	Method of obtaining p-value (Either "liu" or "davies", Default is liu)

**Value**

Score-like test statistic T for the GxE effect and corresponding p-value

**Examples**

```
dat <- makeSimData(H=cosihap, n=500, L=10, gammaG=1, gammaGE=0, causal=4, seed=1)
objSEAGLE <- prep.SEAGLE(y=dat$y, X=dat$X, intercept=1, E=dat$E, G=dat$G)
res <- SEAGLE(objSEAGLE, init.tau=0.5, init.sigma=0.5)
```

---

Vinv

*Function for applying V inverse in Algorithm 1*

---

**Description**

This function applies V inverse via the Woodbury matrix identity

**Usage**

```
Vinv(G, qrM, tau_over_sigma, sigma, RHS)
```

**Arguments**

G	Matrix of genotype markers (size n x L)
qrM	Pre-computation for LxL linear system solve
tau_over_sigma	Tau over sigma from precomputation
sigma	Variance component from model noise epsilon
RHS	Matrix or vector on right-hand side of V inverse

**Value**

Matrix or vector resulting from left multiplication of Vinv with input RHS



# Index

- \* **datasets**
  - cosihap, [3](#)
- applyAt, [2](#)
- cosihap, [3](#)
- estimate.vc, [3](#)
- makeSimData, [4](#)
- prep.SEAGLE, [5](#)
- Rinv.AtG, [6](#)
- Rinv.u, [6](#)
- SEAGLE, [7](#)
- Vinv, [8](#)