

# Package ‘IPLGP’

February 17, 2021

**Type** Package

**Title** Identification of Parental Lines via Genomic Prediction

**Version** 0.1.0

**Description** Combining genomic prediction with Monte Carlo simulation, three different strategies are implemented to select parental lines for multiple traits in plant breeding. The selection strategies include (i) GEBV-O considers only genomic estimated breeding values (GEBVs) of the candidate individuals; (ii) GD-O considers only genomic diversity (GD) of the candidate individuals; and (iii) GEBV-GD considers both GEBV and GD. The above method can be seen in Chung PY, Liao CT (2020) <doi:10.1371/journal.pone.0243159>. Multi-trait genomic best linear unbiased prediction (MT-GBLUP) model is used to simultaneously estimate GEBVs of the target traits, and then a selection index is adopted to evaluate the composite performance of an individual.

**Imports** ggplot2, sommer, grDevices, stats

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**NeedsCompilation** no

**Author** Ping-Yuan Chung [cre],  
Chen-Tuo Liao [aut]

**Maintainer** Ping-Yuan Chung <r06621204@ntu.edu.tw>

**Repository** CRAN

**Date/Publication** 2021-02-17 20:20:06 UTC

## R topics documented:

GA.Dscore . . . . .	2
GBLUP.fit . . . . .	3
output.best . . . . .	4
output.gain . . . . .	5

phe.sd . . . . .	6
simu.gamete . . . . .	7
simu.GDO . . . . .	8
simu.GEBVGD . . . . .	10
simu.GEBVO . . . . .	13

<b>Index</b>	<b>16</b>
--------------	-----------

---

GA.Dscore

*Search For A Subset With The Highest D-score*


---

### Description

Search for an optimal subset of the candidate individuals such that it achieves the highest D-score by genetic algorithm (GA).

### Usage

```
GA.Dscore(
  K,
  size,
  keep = c(),
  n0 = size,
  mut = 3,
  cri = 10000,
  console = FALSE
)
```

### Arguments

K	matrix. An n*n matrix denotes the genomic relationship matrix of the n candidate individuals, where $n > 4$ .
size	integer. An integer denotes the size of the subset, note that $3 < \text{size} < n$ .
keep	vector. A vector indicates those candidate individuals which will be retained in the subset before the search. The length of keep must be less than size.
n0	integer. An integer indicates the number of chromosomes (solutions) in the genetic algorithm, note that $n0 > 3$ .
mut	integer. An integer indicates the number of mutations in the genetic algorithm, note that $\text{mut} < \text{size}$ .
cri	integer. An integer indicates the stopping criterion, note that $\text{cri} < 1e+06$ . The genetic algorithm will stop if the number of iterations reaches cri.
console	logical. A logical variable, if console is set to be TRUE, the searching process will be shown in the R console.

**Value**

subset	The optimal subset with the highest D-score.
D.score	The D.score of the optimal subset.
time	The number of iterations.

**References**

Chung PY, Liao CT. 2020. Identification of superior parental lines for biparental crossing via genomic prediction. PLoS ONE 15(12):e0243159.

Ou JH, Liao CT. 2019. Training set determination for genomic selection. Theor Appl Genet. 132:2781-2792.

**Examples**

```
# generate simulated data
geno.test <- matrix(sample(c(1, -1), 600, replace = TRUE), 20, 30)
K.test <- geno.test%*%t(geno.test)/ncol(geno.test)

# run with no specified individual
result1 <- GA.Dscore(K.test, 6, cri = 1000, console = TRUE)
result1

# run with some specified individuals
result2 <- GA.Dscore(K.test, 6, keep = c(1, 5, 10), cri = 1000, console = TRUE)
result2
```

---

GBLUP.fit

*Muti-trait GBLUP Model*


---

**Description**

Built the multi-trait GBLUP model using the phenotypic and genotypic data of a training population by 'mmer' from R package 'sommer'. Then, output the fitted values of the training population.

**Usage**

```
GBLUP.fit(phe, geno = NULL, K = NULL)
```

**Arguments**

phe	matrix. An n*t matrix contains the phenotypic values of the t target traits.
geno	matrix. An n*p matrix with n individuals and p markers of the training population. The markers must be coded as 1, 0, or -1 for alleles AA, Aa, or aa. The missing value must have been already imputed.
K	matrix. An n*n matrix denotes the genomic relationship matrix of the training population if geno is set to be NULL.

**Value**

The fitted values of the training population.

**References**

Habier D, Fernando RL, Dekkers JCM. 2007. The impact of genetic relationship information on genome-assisted breeding values. *Genetics* 177:2389-2397.

VanRaden PM. 2008. Efficient methods to compute genomic predictions. *J Dairy Sci.* 91:4414-4423.

**See Also**

[mmer](#)

**Examples**

```
# generate simulated data
phe.test <- data.frame(trait1 = rnorm(50,30,10), trait2 = rnorm(50,10,5), trait3 = rnorm(50,20,20))

# run with the marker score matrix
geno.test <- matrix(sample(c(1, -1), 5000, replace = TRUE), 50, 100)
result1 <- GBLUP.fit(phe.test, geno.test)
result1

# run with the genomic relationship matrix
K.test <- geno.test%*%t(geno.test)/ncol(geno.test)
result2 <- GBLUP.fit(phe.test, K = K.test)
result2
```

---

output.best

*Summary For The Best Individuals*

---

**Description**

Output the GEBV average curves and the summary statistics for the best individuals selected over generations.

**Usage**

```
output.best(result, save.pdf = FALSE)
```

**Arguments**

result	list. The data list of the output from simu.GEBVO, simu.GDO, or simu.GEBVGD.
save.pdf	logical. A logical variable, if save.pdf is set to be TRUE, the pdf file of plots will be saved in the working directory instead of being shown in the console.

**Value**

The GEBV averages of the best individuals among the repetitions over generations for each trait.

**Note**

The figure output contains the plots of GEBV averages of the best individuals selected over generations for each trait. If `save.pdf` is set to be `TRUE`, the pdf file of plots will be saved in the working directory instead of being shown in the console.

**References**

Chung PY, Liao CT. 2020. Identification of superior parental lines for biparental crossing via genomic prediction. PLoS ONE 15(12):e0243159.

**See Also**

[simu.GEBVO](#) [simu.GDO](#) [simu.GEBVGD](#) [ggplot](#)

**Examples**

```
# generate simulated data
set.seed(2000)
phe.test <- data.frame(trait1 = rnorm(10,30,10), trait2 = rnorm(10,10,5))
geno.test <- matrix(sample(c(1, -1), 200, replace = TRUE), 10, 20)
marker.test <- cbind(rep(1:2,each=10),rep(seq(0,90,10),2))
geno.candidate <- matrix(sample(c(1,-1),20,replace = TRUE),20,20)

# run
result <- simu.GEBVO(phe.test, geno.test, marker.test, geno.candidate,
nprog = 5, nsele = 10, ngen = 5, nrep = 5)

# summary for the best individuals
output <- output.best(result)
output
```

---

output.gain

*Summary For Genetic Gain*

---

**Description**

Output the GEBV average of parental lines, the GEBV average of the last generation in simulation process, and the genetic gain average over repetitions for each target trait.

**Usage**

```
output.gain(result)
```

**Arguments**

result                    list. The data list of the output from simu.GEBVO, simu.GDO, or simu.GEBVGD.

**Value**

The output contains the table of the GEBV average of parental lines, the GEBV average of the last generation in simulation process, and the genetic gain average over repetitions for each target trait.

**References**

Chung PY, Liao CT. 2020. Identification of superior parental lines for biparental crossing via genomic prediction. PLoS ONE 15(12):e0243159.

**See Also**

[simu.GEBVO](#) [simu.GDO](#) [simu.GEBVGD](#)

**Examples**

```
# generate simulated data
set.seed(2000)
phe.test <- data.frame(trait1 = rnorm(10,30,10), trait2 = rnorm(10,10,5))
geno.test <- matrix(sample(c(1, -1), 200, replace = TRUE), 10, 20)
marker.test <- cbind(rep(1:2,each=10),rep(seq(0,90,10),2))
geno.candidate <- matrix(sample(c(1,-1),20,replace = TRUE),20,20)

# run
result <- simu.GEBVO(phe.test, geno.test, marker.test, geno.candidate,
nprog = 5, nsele = 10, ngen = 5, nrep = 5)

# summary for genetic gain
output <- output.gain(result)
output
```

---

phe.sd

*Standardize Phenotypic Values*

---

**Description**

Standardize the phenotypic values of all the target traits from a training population. Then, output the standardized phenotypic values, the mean vector, and the standard deviation vector of the target traits.

**Usage**

phe.sd(phe)

**Arguments**

`phe` matrix. An  $n \times t$  matrix with  $n$  individuals and  $t$  traits, denotes the phenotypic values. The missing value must be coded as NA.

**Value**

`standardize.phe` An  $n \times t$  matrix contains the standardized phenotypic values.

`mu` A vector with length  $t$  contains the averages of the phenotypic values of the  $t$  target traits.

`sd` A vector with length  $t$  contains the standard deviations of the phenotypic values of the  $t$  target traits.

**Examples**

```
# generate simulated data
phe.test <- data.frame(trait1 = rnorm(50,30,10), trait2 = rnorm(50,10,5), trait3 = rnorm(50,20,20))

# run and output
result <- phe.sd(phe.test)
result
```

---

 simu.gamete

---

*Simulate The Genotypic Values Of A Gamete*


---

**Description**

Generate the genotypic values of a gamete from the genotypic data of its parents by Monte Carlo simulation. The recombination rate is caculate by Haldane's mapping function.

**Usage**

```
simu.gamete(marker)
```

**Arguments**

`marker` data frame. A  $p \times 4$  data frame whose first column indicates the chromosome number to which a marker belongs; second column indicates the position of the marker in centi-Morgan (cM); and 3rd and 4th columns indicates the genotype of the marker (numeric or character).

**Value**

The SNP sequence of gamete.

**References**

Haldane J.B.S. 1919. The combination of linkage values and the calculation of distance between the loci for linked factors. *Genetics* 8: 299–309.

## Examples

```
# generate simulated data
marker.test <- data.frame(c(1,1,1,1,1,2,2,2,2,2),c(10,20,30,40,50,10,20,30,40,50),
c("A","T","C","G","A","A","G","A","T","A"),c("A","A","G","C","T","A","G","T","T","A"))

# run
simu.gamete(marker.test)
```

---

 simu.GDO

*GD-O Strategy*


---

## Description

Identify parental lines based on GD-O strategy and simulate their offsprings.

## Usage

```
simu.GDO(
  phe.t,
  geno.t,
  marker,
  geno.c = NULL,
  npl = NULL,
  better.c = FALSE,
  weight = NULL,
  direction = NULL,
  nprog = 50,
  nsele = NULL,
  ngen = 10,
  nrep = 30,
  cri = 10000,
  console = TRUE
)
```

## Arguments

phe.t	matrix. An n*t matrix denotes the phenotypic values of the training population with n individuals and t target traits.
geno.t	matrix. An n*p matrix denotes the marker score matrix of the training population. The markers must be coded as 1, 0, or -1 for alleles AA, Aa, or aa. The missing value must have been already imputed.
marker	matrix. An p*2 matrix whose first column indicates the chromosome number to which a marker belongs; and second column indicates the position of the marker in centi-Morgan (cM).



<code>geno.c</code>	matrix. An $nc \times p$ matrix denotes the marker score matrix of the candidate population with $nc$ individuals and $p$ markers. The markers must be coded as 1, 0, or -1 for alleles AA, Aa, or aa. The missing value must have been already imputed. If <code>geno.c</code> is set to be NULL, the candidate population is exactly the training population.
<code>np1</code>	integer. An integer indicates the number of individuals who will be chosen as the parental lines. If <code>np1 = NULL</code> , it will be 4 times the number of traits.
<code>better.c</code>	logical. A logical variable, if <code>better.c</code> is set to be TRUE, the candidate individuals with GEBVs better than average for all the target traits will comprise the candidate set. Otherwise, all the candidate individuals will comprise the candidate set.
<code>weight</code>	vector. A vector with length $t$ indicates the weights of target traits in selection index. If <code>weight</code> is set to be NULL, the equal weight will be assigned to all the target traits.
<code>direction</code>	vector. A vector with length $t$ indicates the selecting directions for target traits. The elements of <code>direction</code> are 1, or -1 representing the rule that the larger the better; or the smaller the better. If <code>direction</code> is set to be NULL, the selecting direction will be the same as <code>weight</code> .
<code>nprog</code>	integer. An integer indicates the number of progenies which will be produced for each of the best individuals at every generation.
<code>nsele</code>	integer. An integer indicates the number of the best individuals which will be selected at each generation. If <code>nsele</code> is set to be NULL, the number will be the same as the number of F1 individuals.
<code>ngen</code>	integer. An integer indicates the number of generations in the simulation process.
<code>nrep</code>	integer. An integer indicates the number of repetitions in the simulation process.
<code>cri</code>	integer. An integer indicates the stopping criterion, note that $cri < 1e+06$ . The genetic algorithm will stop if the number of iterations reaches <code>cri</code> .
<code>console</code>	logical. A logical variable, if <code>console</code> is set to be TRUE, the simulation process will be shown in the R console.

### Value

<code>method</code>	The GD-O strategy.
<code>weight</code>	The weights of target traits in selection index.
<code>direction</code>	The selecting directions of target traits in selection index.
<code>mu</code>	The mean vector of target traits.
<code>sd</code>	The standard deviation vector of target traits.
<code>GEBV.value</code>	The GEBVs of target traits in each generation and each repetition.
<code>parental.lines</code>	The IDs and D-score of parental lines selected in each repetition.
<code>suggested.subset</code>	The most frequently selected parental lines by this strategy.

**Note**

The function `output.best` and `output.gain` can be used to summarize the result.

**References**

Chung PY, Liao CT. 2020. Identification of superior parental lines for biparental crossing via genomic prediction. PLoS ONE 15(12):e0243159.

**See Also**

[phe.sd](#) [GBLUP.fit](#) [GA.Dscore](#) [simu.gamete](#) [simu.GEBVO](#) [simu.GEBVGD](#) [output.best](#) [output.gain](#)

**Examples**

```
# generate simulated data
set.seed(2000)
phe.test <- data.frame(trait1 = rnorm(10,30,10), trait2 = rnorm(10,10,5))
geno.test <- matrix(sample(c(1, -1), 200, replace = TRUE), 10, 20)
marker.test <- cbind(rep(1:2, each=10), rep(seq(0, 90, 10), 2))
geno.candidate <- matrix(sample(c(1,-1), 300, replace = TRUE), 15, 20)

# run and output
result <- simu.GDO(phe.test, geno.test, marker.test, geno.candidate,
nprog = 5, nsele = 10, ngen = 5, nrep = 5, cri = 250)
result$suggested.subset
```

---

simu.GEBVGD

*GEBV-GD Strategy*

---

**Description**

Identify parental lines based on GEBV-GD strategy and simulate their offsprings.

**Usage**

```
simu.GEBVGD(
  phe.t,
  geno.t,
  marker,
  geno.c = NULL,
  npl = NULL,
  better.c = FALSE,
  npl.best = NULL,
  weight = NULL,
  direction = NULL,
  nprog = 50,
  nsele = NULL,
  ngen = 10,
```

```

nrep = 30,
cri = 10000,
console = TRUE
)

```

### Arguments

phe.t	matrix. An $n \times t$ matrix denotes the phenotypic values of the training population with $n$ individuals and $t$ target traits.
geno.t	matrix. An $n \times p$ matrix denotes the marker score matrix of the training population. The markers must be coded as 1, 0, or -1 for alleles AA, Aa, or aa. The missing value must have been already imputed.
marker	matrix. An $p \times 2$ matrix whose first column indicates the chromosome number to which a marker belongs; and second column indicates the position of the marker in centi-Morgan (cM).
geno.c	matrix. An $nc \times p$ matrix denotes the marker score matrix of the candidate population with $nc$ individuals and $p$ markers. The markers must be coded as 1, 0, or -1 for alleles AA, Aa, or aa. The missing value must have been already imputed. If <code>geno.c</code> is set to be NULL, the candidate population is exactly the training population.
npl	integer. An integer indicates the number of individuals who will be chosen as the parental lines. If <code>npl = NULL</code> , it will be 4 times the number of traits.
better.c	logical. A logical variable, if <code>better.c</code> is set to be TRUE, the candidate individuals with GEBVs better than average for all the target traits will comprise the candidate set. Otherwise, all the candidate individuals will comprise the candidate set.
npl.best	vector. A vector indicates the numbers of the best candidate individuals which will be retained for each target trait before the search. If <code>npl.best</code> is set to be NULL, there will be 2 best candidate individuals retained for each trait.
weight	vector. A vector with length $t$ indicates the weights of target traits in selection index. If <code>weight</code> is set to be NULL, the equal weight will be assigned to all the target traits.
direction	vector. A vector with length $t$ indicates the selecting directions for target traits. The elements of <code>direction</code> are 1, or -1 representing the rule that the larger the better; or the smaller the better. If <code>direction</code> is set to be NULL, the selecting direction will be the same as <code>weight</code> .
nprog	integer. An integer indicates the number of progenies which will be produced for each of the best individuals at every generation.
nsele	integer. An integer indicates the number of the best individuals which will be selected at each generation. If <code>nsele</code> is set to be NULL, the number will be the same as the number of F1 individuals.
ngen	integer. An integer indicates the number of generations in the simulation process.
nrep	integer. An integer indicates the number of repetitions in the simulation process.

<code>cri</code>	integer. An integer indicates the stopping criterion, note that $cri < 1e+06$ . The genetic algorithm will stop if the number of iterations reaches <code>cri</code> .
<code>console</code>	logical. A logical variable, if <code>console</code> is set to be <code>TRUE</code> , the simulation process will be shown in the R console.

**Value**

<code>method</code>	The GEBV-GD strategy.
<code>weight</code>	The weights of target traits in selection index.
<code>direction</code>	The selecting directions of target traits in selection index.
<code>mu</code>	The mean vector of target traits.
<code>sd</code>	The standard deviation vector of target traits.
<code>GEBV.value</code>	The GEBVs of target traits in each generation and each repetition.
<code>parental.lines</code>	The IDs and D-score of parental lines selected in each repetition.
<code>suggested.subset</code>	The most frequently selected parental lines by this strategy.

**Note**

The function `output.best` and `output.gain` can be used to summarize the result.

**References**

Chung PY, Liao CT. 2020. Identification of superior parental lines for biparental crossing via genomic prediction. *PLoS ONE* 15(12):e0243159.

**See Also**

[phe.sd](#) [GBLUP.fit](#) [GA.Dscore](#) [simu.gamete](#) [simu.GEBVO](#) [simu.GDO](#) [output.best](#) [output.gain](#)

**Examples**

```
# generate simulated data
set.seed(2000)
phe.test <- data.frame(trait1 = rnorm(10,30,10), trait2 = rnorm(10,10,5))
geno.test <- matrix(sample(c(1, -1), 200, replace = TRUE), 10, 20)
marker.test <- cbind(rep(1:2, each=10), rep(seq(0, 90, 10), 2))
geno.candidate <- matrix(sample(c(1,-1), 300, replace = TRUE), 15, 20)

# run and output
result <- simu.GEBVGD(phe.test, geno.test, marker.test, geno.candidate,
nprog = 5, nsele = 10, ngen = 5, nrep = 5, cri = 250)
result$suggested.subset
```

simu.GEBVO

*GEBV-O Strategy***Description**

Identify parental lines based on GEBV-O strategy and simulate their offsprings.

**Usage**

```
simu.GEBVO(
  phe.t,
  geno.t,
  marker,
  geno.c = NULL,
  npl = NULL,
  weight = NULL,
  direction = NULL,
  nprog = 50,
  nsele = NULL,
  ngen = 10,
  nrep = 30,
  console = TRUE
)
```

**Arguments**

<code>phe.t</code>	matrix. An $n \times t$ matrix denotes the phenotypic values of the training population with $n$ individuals and $t$ target traits.
<code>geno.t</code>	matrix. An $n \times p$ matrix denotes the marker score matrix of the training population. The markers must be coded as 1, 0, or -1 for alleles AA, Aa, or aa. The missing value must have been already imputed.
<code>marker</code>	matrix. An $p \times 2$ matrix whose first column indicates the chromosome number to which a marker belongs; and second column indicates the position of the marker in centi-Morgan (cM).
<code>geno.c</code>	matrix. An $nc \times p$ matrix denotes the marker score matrix of the candidate population with $nc$ individuals and $p$ markers. The markers must be coded as 1, 0, or -1 for alleles AA, Aa, or aa. The missing value must have been already imputed. If <code>geno.c</code> is set to be NULL, the candidate population is exactly the training population.
<code>npl</code>	vector. A vector denotes how many parental lines with the top GEBVs will be chosen from each trait. If <code>npl</code> is set to be NULL, there will be 4 parental lines chosen from each trait.
<code>weight</code>	vector. A vector with length $t$ indicates the weights of target traits in selection index. If <code>weight</code> is set to be NULL, the equal weight will be assigned to all the target traits.

direction	vector. A vector with length t indicates the selecting directions for target traits. The elements of direction are 1, or -1 representing the rule that the larger the better; or the smaller the better. If direction is set to be NULL, the selecting direction will be the same as weight.
nprog	integer. An integer indicates the number of progenies which will be produced for each of the best individuals at every generation.
nsele	integer. An integer indicates the number of the best individuals which will be selected at each generation. If nsele is set to be NULL, the number will be the same as the number of F1 individuals.
ngen	integer. An integer indicates the number of generations in the simulation process.
nrep	integer. An integer indicates the number of repetitions in the simulation process.
console	logical. A logical variable, if console is set to be TRUE, the simulation process will be shown in the R console.

### Value

method	The GEBV-O strategy.
weight	The weights of target traits in selection index.
direction	The selecting directions of target traits in selection index.
mu	The mean vector of target traits.
sd	The standard deviation vector of target traits.
GEBV.value	The GEBVs of target traits in each generation and each repetition.
parental.lines	The IDs and D-score of parental lines selected in each repetition.
suggested.subset	The most frequently selected parental lines by this strategy.

### Note

The function `output.best` and `output.gain` can be used to summarize the result.

### References

Chung PY, Liao CT. 2020. Identification of superior parental lines for biparental crossing via genomic prediction. PLoS ONE 15(12):e0243159.

### See Also

[phe.sd](#) [GBLUP.fit](#) [GA.Dscore](#) [simu.gamete](#) [simu.GDO](#) [simu.GEBVGD](#) [output.best](#) [output.gain](#)

### Examples

```
# generate simulated data
set.seed(2000)
phe.test <- data.frame(trait1 = rnorm(10,30,10), trait2 = rnorm(10,10,5))
geno.test <- matrix(sample(c(1, -1), 200, replace = TRUE), 10, 20)
```

```
marker.test <- cbind(rep(1:2, each=10), rep(seq(0, 90, 10), 2))
geno.candidate <- matrix(sample(c(1,-1), 300, replace = TRUE), 15, 20)

# run and output
result <- simu.GEBVO(phe.test, geno.test, marker.test, geno.candidate,
nprog = 5, nsele = 10, ngen = 5, nrep = 5)
result$suggested.subset
```

# Index

GA.Dscore, [2](#), [10](#), [12](#), [14](#)

GBLUP.fit, [3](#), [10](#), [12](#), [14](#)

ggplot, [5](#)

mmer, [4](#)

output.best, [4](#), [10](#), [12](#), [14](#)

output.gain, [5](#), [10](#), [12](#), [14](#)

phe.sd, [6](#), [10](#), [12](#), [14](#)

simu.gamete, [7](#), [10](#), [12](#), [14](#)

simu.GDO, [5](#), [6](#), [8](#), [12](#), [14](#)

simu.GEBVGD, [5](#), [6](#), [10](#), [10](#), [14](#)

simu.GEBVO, [5](#), [6](#), [10](#), [12](#), [13](#)