HIBLUP User Manual

September 24, 2018
v1.01

Disclaimer: While extensive tests have been performed by Zhao lab at Huazhong Agricultural University and Yuan lab at Wuhan University of Technology. Results are, in general, reliable, correct, and appropriate. However, results are not guaranteed for any specific data set. We strongly recommend that users validate the HIBLUP results with other software packages, such as GCTA, LDAK, and DMU.

Support documents: Extensive support documents, including the user manual, demo script, demo data and demo results, are available at the zip file.

Questions and comments: Users and developers are recommended to send questions to Lilin Yin (ylilin@163.com), Haohao Zhang (haohaozhang@whut.edu.cn), and Xiaolei Liu (xiaoleiliu@mail.hzau.edu.cn).

Authors: Lilin Yin[†], Haohao Zhang[†], Zhenshuang Tang, Dong Yin, Xinyun Li, Xiaohui Yuan, Shuhong Zhao, and **Xiaolei Liu**.

[†]These authors contributed equally to this work.

Contents

1 Installing HIBLUP and a quick start

1.1 Installation

It is highly recommended to install Microsoft R Open (`https://mran.microsoft.com/download/`) to speed up the mathematical calculation of HIBLUP, but this is not required, and HIBLUP can also work with base R. HIBLUP can be easily installed using the following codes:

```
install.packages("Rcpp")
install.packages("RcppParallel")
install.packages("RcppArmadillo")
install.packages("bigmemory")
install.packages("hiblup_1.0.1.tar.gz", repos=NULL)
```

1.2 Quick start

The data embedded in HIBLUP was derived from an animal breeding farm, it includes a total of 2934 genetic related individuals and 573 of them were genotyped with 50K SNP Chip. The genotype was coded as 0, 1, 2 for AA, AB, BB, respectively, and two traits(t1, t2) were recorded for 800 individuals. Sire information and sex information can be treated as random effect and fixed effect, respectively. A quick start of HIBLUP to fit above model is shown below:

```
library("hiblup")
data("hidata")
X <- model.matrix(~Sex, data=pheno)  # fixed effects
R <- as.matrix(pheno$Sire)           # random effects

gebv <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map, geno.id=geno.id,
    pedigree=pedigree, vc.method=c("HI"), mode="A", CV=X, R=R,
    back.solution=TRUE)
```

```
SSBLUP model is selected based on provided data
Analyzed trait: t1
Number of covariate: 3
Number of random: 1
Number of individuals with phenotype: 800
Deriving GA matrix from genotype... Done within 1s
Number of genotyped individuals: 573
Number of genotyped individuals with phenotype: 175
Number of genotyped individuals without phenotype: 398
Deriving A matrix from pedigree...Done within 0s
Number of total predicted individuals: 2934
Realign index of y...Done!
Realign index of X matrix...Done!
Realign index of R matrix...Done!
Extracting A11 matrix...Done!
```

```
Mean of diagonal and Off-diagonal of PA: 1.0002 0.0248
Mean of diagonal and Off-diagonal of GA: 0.9886 -0.0017
Adjusting GA matrix: GA* = 0.99 * GA + 0.03
Weighting of A11 and GA matrix: 0.05
Calculating inverse of A11 matrix...Done within 1s
Constructing HA matrix...Done within 1s
HE Prior derived: A:0.1 e:89.08749; Done within 0s
HE adopted: TRUE
Variance components estimation:
[Iter]  Var_R1(SE)    Var_K1(SE)      Var_e(SE)      h2_R1(SE)      h2_K1(SE)
[AI] 13.958(15.0571) 1.798(7.9697) 87.218(7.4915) 0.1356(0.1270) 0.0175(0.0772)
[AI] 0.624(6.4713) 6.624(8.4325) 81.838(7.6181) 0.0070(0.0722) 0.0744(0.0934)
[AI] 2.758(0.8631) 3.910(8.5912) 84.307(7.3534) 0.0303(0.0100) 0.0430(0.0934)
[AI] 4.684(1.5499) 4.511(8.3950) 83.587(7.4085) 0.0505(0.0167) 0.0486(0.0894)
[AI] 5.613(2.2921) 5.434(8.8501) 82.817(7.6245) 0.0598(0.0238) 0.0579(0.0930)
[AI] 5.803(2.6863) 5.881(9.2503) 82.468(7.8018) 0.0616(0.0276) 0.0625(0.0967)
[AI] 5.827(2.7739) 5.994(9.4167) 82.381(7.8736) 0.0619(0.0284) 0.0636(0.0984)
[AI] 5.830(2.7858) 6.013(9.4563) 82.366(7.8904) 0.0619(0.0285) 0.0638(0.0988)
[AI] 5.831(2.7877) 6.016(9.4630) 82.363(7.8933) 0.0619(0.0286) 0.0639(0.0989)
[AI] 5.831(2.7881) 6.017(9.4641) 82.363(7.8938) 0.0619(0.0286) 0.0639(0.0989)
[AI] 5.831(2.7881) 6.017(9.4643) 82.363(7.8939) 0.0619(0.0286) 0.0639(0.0989)
[AI] 5.831(2.7881) 6.017(9.4644) 82.363(7.8939) 0.0619(0.0286) 0.0639(0.0989)
[Convergence] YES
Done within 5s
Estimated beta: 149.7262 4.366075
Estimated Vg and Ve: 6.017368 82.36306
HIBLUP DONE WITHIN TOTAL RUN TIME: 9s
HIBLUP ACCOMPLISHED SUCCESSFULLY!
```

You can also load your own data with the following codes:

```
pheno <- read.table("phenotype.txt", header=F)
geno <- bigmemory::attach.big.matrix("genotype.desc")
geno.id <- read.table("geno.id", header=F)
pedigree <- read.table("pedigree.txt", header=T)
map <- read.table("map.txt", header=F)
```

Note that two result files will be generated. One is used for storing the estimated genetic values, and the other is used for storing marker effects if "back.solution" is TRUE. The contents of these two files are displayed as follows:

Ind	hiblup.A.ebv	hiblup.D.ebv	hiblup.AD.ebv
P0322	-0.0805	-3.21e-04	-0.0808
P0323	-0.1635	-3.14e-04	-0.1638
P0324	-0.0830	1.04e-06	-0.0830
P0325	-0.0637	1.31e-06	-0.0636

Table 1: Estimated genetic values.

Marker	Chr	Pos	P.Freq	SNP.A.effect	SNP.D.effect
ASGA0000014	1	342481	0.355	-0.0304	0.01134
ASGA0000021	1	489855	0.407	-0.0473	-0.00152
H3GA0000026	1	509928	0.286	0.2445	0.09484
ALGA0000009	1	538161	0.139	-0.2514	-0.07195
ALGA0000014	1	565627	0.390	-0.1583	-0.00737

Table 2: Genetic marker effects.

2 Input of HIBLUP

The data requirements of three BLUP methods in HIBLUP:
ABLUP: Phenotypic observations, Pedigree records
GBLUP: Phenotypic observations, Genotype data
SSBLUP: Phenotypic observations, Genotype data, and Pedigree records

2.1 Pedigree Data

The pedigree data file includes 3 columns (sample id, paternal id, and maternal id). Note that the individuals in the pedigree data file do not need to be sorted by the date of birth, and the missing value can be replaced by NA or 0.

```
R> pedigree[1:3, ]
```

```
    ID Sire  Dam
1 ind1 <NA> <NA>
2 ind2 <NA> <NA>
3 ind3 <NA> <NA>
```

2.2 Genotype data

HIBLUP accepts both "big.matrix" format, which is from R bigmemory package and R standard "matrix" format. Each Column represents an individual and each row represents a marker. Here is an example that contains 573 individuals and each individual has 48,353 markers from the demo data. Genotype data in multiple popular formats such as vcf, hapmap, and plink binary format can be converted to "big.matrix" using "MVP.Data" function in the MVP package (`https://github.com/XiaoleiLiuBio/MVP`). Genotype ID list is a one-column matrix that includes the id list of genotyped individusls. The order of individuals in genotype id list should match the order of individuals in Genotype data file.

```
R> geno.id[1:3, ]
```

```
[1] ind799 ind800 ind801
573 Levels: ind1061 ind1063 ind1066 ind1067 ind1068 ... ind842
```

```
R> dim(geno)
```

```
[1] 48353   573
```

```
R> geno[1:3, 1:10]
```

```
  V1 V2 V3 V4 V5 V6 V7 V8 V9 V10
1  1  1  2  0  1  2  1  1  0   1
2  1  1  2  2  2  0  2  1  2   1
3  1  1  1  1  1  0  0  1  2   0
```

2.3 Genotypic map data

Genotypic map data includes three columns, which are marker id, Chromosome ID, and physical position. This information is only used for the output.

```
R> dim(map)
```

```
[1] 48353     3
```

```
R> map[1:3, ]
```

```
                  SNP Chrom      BP
1           1_242598     1  242598
2   WU_10.2_1_266158     1  266158
3        ASGA0000014     1  342481
```

2.4 Phenotype, Fixed effects, and Random effects

Individuals in Phenotype, fixed effects, and random effects must have the same order and the individual ID is only added in the first column of phenotype data. Above three information are always incorporated in a single file. Missing phenotype value should be marked with "NA".

```
R> dim(pheno)
```

```
[1] 800   5
```

```
R> pheno[1:3, ]
```

```
       ID   Sire Sex     t1    t2
1 ind2124 ind852   2 156.96 12.31
2 ind1107 ind855   2 153.93 12.87
3 ind1814 ind849   2 154.04 13.33
```

2.5 Variance components

For single trait and K model, `vc=c(`V_g, V_e`)`;
for pairs of correlated traits, `vc=c(`$V_g^{(1)}, V_g^{(2)}, COV_g^{(12)}, V_e^{(1)}, V_e^{(2)}, COV_e^{(12)}$`)`;
for multiple K model, `vc=c(`$V_g^{(1)}, V_g^{(2)} \ldots V_g^{(n)}, V_e$`)`;
if R (Random effects) is added in the model, V_r should be added in the beginning of `vc` vector.

3 Gallery of HIBLUP input parameters

Parameter	Default	Options	Description
Pheno	NULL	Users	Phenotypic observations
CV	NULL	Users	Fixed effects
R	NULL	Users	Random effects
geno	NULL	Users	Genotype data
pedigree	NULL	Users	Pedigree records
map	NULL	Users	Genotypic map
geno.id	NULL	Users	Genotype id list
val.id	NULL	Users	sample id list for prediction on individual genetic value
K	NULL	Users	a list of variance-covariance matrices for random effects
G	NULL	Users	Relationship matrix that derived from genotype data
A	NULL	Users	Relationship matrix that derived from pedigree records
alpha	0.05	$0 \sim 1$	the weight of A matrix when merging A and G matrices
cpu	NULL	Positive integer	number of threads used for parallel computation, default is NULL and automatically assign the computational task to appropriate number of threads
vc	NULL	Users	A vector includes known variance components. See section 2.5
mode	"A"	"A" or "AD"	"A" and "AD" repsent Additive model and Additive plus Dominant model, respectively; it doesn't work when "K" is not NULL
vc.method	HI	"AI", "EM", "AIEM", "EMAI", "HE", and "HI"	methods for variance components estimation
nAIiter	20	Positive integer	Maximum iteration number for "AI" algorithm
nEMiter	1	Positive integer	Maximum iteration number for "EM" algorithm
mme.method	"sor"	"solve" and "sor"	methods for solving mixed model equation when vc is known
reliability	FALSE	TRUE or FALSE	if TRUE, the reliability of individual genetic value will be calculated
back.solution	FALSE	TRUE or FALSE	if TRUE, the marker effects will be calculated
file.output	TRUE	TRUE or FALSE	if TRUE, gebv and marker effect will be written out
het.add	FALSE	TRUE or FALSE	if TRUE, the individual heterozygosity will be added as covariates in AD model

Table 3: Gallery of HIBLUP input parameters

4 Functions and scripts

In this section, we will provide some code snippets to show the HIBLUP functions. For the sake of brevity, output has been hidden. All code has been verified under the built-in data set.

```
data(hidata)
```

4.1 Load data

You can also load your own data with the following codes:

```
pheno <- read.table("phenotype.txt", header=F)
geno <- bigmemory::attach.big.matrix("genotype.desc")
geno.id <- read.table("geno.id", header=F)
pedigree <- read.table("pedigree.txt", header=T)
map <- read.table("map.txt", header=F)
```

4.2 Construct relationship matrix

4.2.1 Pedigree based relationship matrix(A matrix)

Construct pedigree based Additive relationship matrix:

```
Acal <- hiblup.AD(pedigree=pedigree, mode="A")
Amat <- Acal$PA
id <- Acal$order.id
```

Construct pedigree based Additive and Dominant relationship matrix:

```
ADcal <- hiblup.AD(pedigree=pedigree, mode="AD")
Amat <- ADcal$PA
Dmat <- ADcal$PD
id <- ADcal$order.id
```

4.2.2 Genome based relationship matrix(G matrix)

Construct genome based Additive relationship matrix:

```
Acal <- hiblup.K(M=geno, mode="A")
KA <- Acal$GA
```

Construct genome based Additive and Dominant relationship matrix:

```
ADcal <- hiblup.K(M=geno, mode="AD")
KA <- ADcal$GA
KD <- ADcal$GD
```

4.2.3 Pedigree and genome based relationship matrix(H matrix)

Construct pedigree and genome based Additive relationship matrix:

```
G_ind <- as.character(as.matrix(geno.id)[, 1])
phe_ind <- as.character(as.matrix(pheno)[, 1])

Acal <- hiblup.AD(pedigree=pedigree, mode="A")
PA <- Acal$PA
A_ind <- Acal$order.id

Acal <- hiblup.AD(pedigree, mode="A", inverse=TRUE)
PAinv <- Acal$PA

K <- hiblup.K(M=geno, mode="A")
GA <- K$GA

H <- hiblup.H(A_ind=A_ind, G_ind=G_ind, phe_ind=phe_ind, A=PA,
    G=GA, Ainv=PAinv, alpha=0.05, tag="a")
```

Construct pedigree and genome based Additive and Dominant relationship matrix:

```
G_ind <- as.character(as.matrix(geno.id)[, 1])
phe_ind <- as.character(as.matrix(pheno)[, 1])

AD <- hiblup.AD(pedigree=pedigree, mode="AD")
PA <- AD$PA
PD <- AD$PD
A_ind <- Acal$order.id

AD <- hiblup.AD(pedigree, mode="AD", inverse=TRUE)
PAinv <- AD$PA
PDinv <- AD$PD

K <- hiblup.K(M=geno, mode="AD")
GA <- K$GA
GD <- K$GD

HA <- hiblup.H(A_ind=A_ind, G_ind=G_ind, phe_ind=phe_ind, A=PA,
    G=GA, Ainv=PAinv, alpha=0.05, tag="a")
HD <- hiblup.H(A_ind=A_ind, G_ind=G_ind, phe_ind=phe_ind, A=PD,
    G=GD, Ainv=PDinv, alpha=0.05, tag="d")
```

4.3 Variance components estimation

Six variance components estimation methods were implemented in HIBLUP, including AI, EM, EMAI, AIEM, HE Regression, and HI. Since the information required for HE Regression

is different from other methods, it is implemented separately in a function named `hiblup.he`. Other methods can be called by setting the `method` parameter of the `hiblup.vc` function. `nAIiter` and `nEMiter` are the maximum iteration number of "AI" and "EM", valid only in the variance components estimation with "AI" or "EM" method.

4.3.1 Singe K model

```
index <- match(geno.id[, 1], pheno[, 1])

# AI
vc <- hiblup.vc(y=pheno$t1[index], K=KA, nAIiter=20, method="AI")

# EM
vc <- hiblup.vc(y=pheno$t1[index], K=KA, nEMiter=20, method="EM")

# HI
vc <- hiblup.vc(y=pheno$t1[index], K=KA, nAIiter=20, method="HI")

# EMAI
vc <- hiblup.vc(y=pheno$t1[index], K=KA, nAIiter=20, nEMiter=1,
    method="EMAI")

# AIEM
vc <- hiblup.vc(y=pheno$t1[index], K=KA, nAIiter=5, nEMiter=20,
    method="AIEM")

# HE Regression
vc <- hiblup.he(y=pheno$t1[index], K=KA)
```

Fixed effects and random effects can be added by parameters `X` and `R`:

```
X <- model.matrix(~Sex, data=pheno)  # fixed effects
R <- as.matrix(pheno$Sire)           # random effects

vc <- hiblup.vc(y=pheno$t1[index], X=X[index, ], R=R[index, ], K=KA,
    nAIiter=20, method="AI")
```

4.3.2 Multiple K model

Parameter `K` accepts a list of `K`s and execute a multiple random effects model:

```
index <- match(geno.id[, 1], pheno[, 1])

ADcal <- hiblup.K(M=geno, mode="AD")
KA <- ADcal$GA
KD <- ADcal$GD

# AI, EM, EMAI, AIEM, and HI algorithm
vc <- hiblup.vc(y=pheno$t1[index], K=list(KA,KD), method="AI")
# HE Regression
vc <- hiblup.he(y=pheno$t1[index], K=list(KA,KD))
```

With fixed effects and random effects:

```
index <- match(geno.id[, 1], pheno[, 1])

ADcal <- hiblup.K(M=geno, mode="AD")
KA <- ADcal$GA
KD <- ADcal$GD

X <- model.matrix(~Sex, data=pheno)  # fixed effects
R <- as.matrix(pheno$Sire)           # random effects

# AI, EM, EMAI, AIEM, and HI algorithm
vc <- hiblup.vc(y=pheno$t1[index], X=X[index, ], R=R[index, ],
    K=list(KA,KD), method="AI")

# HE algorithm
vc <- hiblup.he(y=pheno$t1[index], X=X[index, ], R=R[index, ],
    K=list(KA,KD))
```

4.3.3 Pairs of correlated traits

Variance and co-variance for pairs of correlated traits can be estimated using following codes. X1 and X2 are the fixed effects of trait1 and trait2, respectively.

```
X <- model.matrix(~Sex, data=pheno)  # fixed effects

vc <- hiblup.bivar.vc(y1=pheno$t1, y2=pheno$t2, X1=X, X2=X,
    K=Amat[1:nrow(pheno), 1:nrow(pheno)], method="AI")
```

4.3.4 With user-provided variance components

The `start` parameter is used to accept the initial value in the variance component calculation method containing AI or EM. The length of the start vector is equal to the number of K plus one. It should be noted that if R is specified, the value of V_R needs to be given in the first position in start. For pairs of correlated traits, the elements in start are $V_g^{(1)}$, $V_g^{(2)}$, $COV_g^{(12)}$, $V_e^{(1)}$, $V_e^{(2)}$, and $COV_e^{(12)}$;

```
# Single K model
start <- c(7.650, 102.788)
vc <- hiblup.vc(y=pheno$t1[index], K=KA, start=start, method="AI")

# Single K model with fixed effects and random effects
start <- c(6.166, 1.137, 93.372)
vc <- hiblup.vc(y=pheno$t1[index], X=X[index, ], R=R[index, ], K=KA,
    start=start, method="AI")

# Multiple K model
start <- c(0.002, 18.804, 92.654)
vc <- hiblup.vc(y=pheno$t1[index], K=list(KA,KD), start=start,
    method="AI")

# Multiple K model with fixed effects and random effects
start <- c(31.409, 0.011, 21.212, 57.137)
vc <- hiblup.vc(y=pheno$t1[index], X=X[index, ], R=R[index, ],
    start=start, K=list(KA,KD), method="AI")

# Pairs of correlated traits
start <- c(39.811, 0.002, -41.416, 78.692, 2.70, 13.684)
vc <- hiblup.bivar.vc(y1=pheno$t1, y2=pheno$t2, X1=X, X2=X, start=start,
    K=Amat[1:nrow(pheno), 1:nrow(pheno)], method="AI")
```

4.4 BLUP

4.4.1 Additive effect based model

if variance components are unknown:

```
# AI, EM, EMAI, AIEM, and HI algorithm
X <- model.matrix(~Sex, data=pheno)  # fixed effects
R <- as.matrix(pheno$Sire)           # random effects

# using HI algorithm if variance components are unknown
gebv.a.hi <- hiblup(pheno=pheno[,c(1,4)], pedigree=pedigree, CV=X, R=R,
    vc.method=c("HI"), nAIiter=5, mode="A")
```

4.4.2 Additive and Dominant effect based model

```
gebv.ad <- hiblup(pheno=pheno[,c(1,4)], pedigree=pedigree, mode="AD")
```

4.4.3 With user-provided variance components

If the variance components are known and provided by the users, the methods for solving mixed model equation can be controlled by the `mme.method` parameter. The options are

"solve" and "sor".

```
# Solve mixed model equation directly
gebv.a <- hiblup(pheno=pheno[,c(1,4)], pedigree=pedigree, CV=X, R=R,
    vc=c(57.8930, 0.0686, 0.0008), mme.method="solve", mode="A")

# Solve mixed model equation using SOR method
gebv.a <- hiblup(pheno=pheno[,c(1,4)], pedigree=pedigree, CV=X, R=R,
    vc=c(57.8930, 0.0686, 0.0008), mme.method="sor", mode="A")
```

4.4.4 Pairs of correlated traits

HIBLUP supports the estimation of individual genetic values for pairs of correlated traits. Users can specify the columns of the trait1 and trait2 in phenotype file by setting the bivar.pos parameter, for example:

```
gebv <- hiblup(pheno=pheno, bivar.pos=c(3,4), X1=X, X2=X,
    pedigree=pedigree)
```

4.4.5 Reliability of individual genetic value

The boolean parameter `reliability` is used to specify whether to calculate the reliability of each individual's genetic value.

```
X <- model.matrix(~Sex, data=pheno)  # fixed effects
R <- as.matrix(pheno$Sire)           # random effects

# get the reliability of individual genetic value
gebv.a.hi <- hiblup(pheno=pheno[,c(1,4)], pedigree=pedigree, CV=X, R=R,
    vc.method=c("HI"), nAIiter=5, mode="A", reliability=TRUE)
```

4.5 GBLUP

4.5.1 Additive effect based model

```
# AI, EM, EMAI, AIEM, and HI algorithm
X <- model.matrix(~Sex, data=pheno)   # fixed effects

gebv.a.ai <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map,
    geno.id=geno.id, CV=X, vc.method=c("AI"), mode="A")
```

4.5.2 Additive and Dominant effect based model

```
gebv.ad <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map,
    geno.id=geno.id, mode="AD")
```

4.5.3 With user-provided variance components

If the variance components are known and provided by the users, the methods for solving mixed model equation can be controlled by the `mme.method` parameter. The options are "solve" and "sor".

```
# Solve mixed model equation directly
gebv.a <- hiblup(pheno=pheno[,c(1,4)], mme.method="solve", CV=X, R=R,
    vc=c(57.8930, 0.0686, 0.0008), geno=geno, map=map, geno.id=geno.id)

# Solve mixed model equation using SOR method
gebv.a <- hiblup(pheno=pheno[,c(1,4)], mme.method="sor", CV=X, R=R,
    vc=c(57.8930, 0.0686, 0.0008), geno=geno, map=map, geno.id=geno.id)
```

4.5.4 Estimate the marker effects

HIBLUP will output the marker effects if `back.solution` is `TRUE`. it only works with GBLUP model or SSBLUP model.

```
gebv.a.ai <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map,
    geno.id=geno.id, CV=X, vc.method=c("AI"), mode="A",
    back.solution=TRUE)
```

4.5.5 Pairs of correlated traits

HIBLUP supports the estimation of individual genetic values for pairs of correlated traits. Users can specify the position of the trait1 and trait2 in pheno by setting the `bivar.pos` parameter, for example:

```
gebv <- hiblup(pheno=pheno, bivar.pos=c(3,4), X1=X, X2=X,
    map=map, geno=geno, geno.id=geno.id)
```

4.5.6 Reliability of individual genetic value

The boolean parameter `reliability` is used to specify whether to calculate the reliability of each individual's genetic value.

```
X <- model.matrix(~Sex, data=pheno)   # fixed effects

gebv.a.ai <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map,
    geno.id=geno.id, CV=X, vc.method=c("AI"), mode="A",
    reliability=TRUE)
```

4.6 SSBLUP

4.6.1 Additive effect based model

```
gebv.a.ai <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map,
    geno.id=geno.id, pedigree=pedigree, vc.method=c("AI"), mode="A")
```

4.6.2 Additive and Dominant effect based model

```
gebv.ad <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map,
    geno.id=geno.id, pedigree=pedigree, mode="AD")
```

4.6.3 With user-provided variance components

If the variance components are known and provided by the users, the methods for solving mixed model equation can be controlled by the `mme.method` parameter. The options are "solve" and "sor".

```
# Solve mixed model equation directly
gebv.a <- hiblup(pheno=pheno[,c(1,4)], mme.method="solve", CV=X, R=R,
    vc=c(57.8930, 0.0686, 0.0008), geno=geno, map=map, geno.id=geno.id,
    pedigree=pedigree)

# Solve mixed model equation using SOR method
gebv.a <- hiblup(pheno=pheno[,c(1,4)], mme.method="sor", CV=X, R=R,
    vc=c(57.8930, 0.0686, 0.0008), geno=geno, map=map, geno.id=geno.id,
    pedigree=pedigree)
```

4.6.4 Estimate the marker effects

```
gebv.a.ai <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map,
    geno.id=geno.id, pedigree=pedigree, vc.method=c("AI"), mode="A",
    back.solution=TRUE)
```

4.6.5 Pairs of correlated traits

HIBLUP also supports the estimation of individual genetic values for pairs of correlated traits. Users can specify the position of the trait1 and trait2 in pheno by setting the `bivar.pos` parameter, for example:

```
gebv <- hiblup(pheno=pheno, bivar.pos=c(3,4), X1=X, X2=X,
    pedigree=pedigree, map=map, geno=geno, geno.id=geno.id)
```

4.6.6 Reliability of individual genetic value

The boolean parameter `reliability` is used to specify whether to calculate the reliability of each individual's genetic value.

```
gebv.a.ai <- hiblup(pheno=pheno[,c(1,4)], geno=geno, map=map,
    geno.id=geno.id, pedigree=pedigree, vc.method=c("AI"), mode="A",
    reliability=TRUE)
```

5 Function support list of HIBLUP

		HIBLUP
Input	Genotype	√
	Pedigree	√
	Phenotype	√
	Fixed effects	√
	Random effects	√
	Relationship matrix	√
VC	AI	√
	EM	√
	EMAI	√
	HE Regression	√
	HI	√
Variable	Fixed effects	√
	Random effects	√
Model	ABLUP	√
	GBLUP	√
	SSBLUP	√
Output	GEBV	√
	Effect	√
	Reliability	√

Table 4: Function support list of HIBLUP.

6 HIBLUP Biography

Date	Version	Event
	1.0	
	1.01	