EPIHAP USER MANUAL

VERSION 1.0

MARCH 25, 2025

1 Introduction

Heritability estimation and genomic prediction are critical components of quantitative genetic studies. However, there are limited tools available for working with multifactorial models that may include any or all types of the following effects: single SNP additive and dominance effects, epistatic effects, or haplotype additive effects. To address this problem, here we provide a very helpful program called EPIHAP, which integrates these effects and allows the user to model abundance in the heritability estimation and genomic prediction. This document provides detailed instructions on how to run this program.

EPIHAP is designed for bi-allelic of diploid species, and is based on the genomic best linear unbiased prediction (GBLUP) model, which can incorporate these effects while controlling for fixed non-genetic factors (Da et al., 2022). Notably, as a key feature of our program, users can select up to third-order epistatic effects to thoroughly investigate the potential genetic mechanisms among SNPs. Analogous to the program GVCHAP (Prakapenka et al., 2020), EPIHAP is also very flexible for the users to determine any or all kinds of effects to be included in the models.

To reduce computing time, the first step we recommend is to construct all types of genetic relationship matrices (GRMs), including additive (A), dominance (D), additive × additive (AA), additive × dominance (AD), dominance × dominance (DD), additive × additive × additive (AAA), additive × additive × dominance (AAD), additive × dominance × dominance (ADD), dominance × dominance × dominance (DDD) or haplotype additive (AH) effects. EPIHAP offers two methods for inferring epistatic genomic relationship matrices (GRMs): one is the Approximate Genomic Epistasis Relationship Matrices (AGERM), and the other is the Exact Genomic Epistasis Relationship Matrices (EGERM) (Henderson, 1985; Jiang and Reif, 2020). The default method is AGERM.

Next, given the property model parameters, EPIHAP will use a linear mixed model, GREML_CE, to estimate the variance component and heritability for any or all types of effects, and to compute the genetic values and their reliability values for such effects (Wang et al., 2014).

If the user expects to get not only those values described above but also the effects and heritability estimates for each SNP, each pair of SNPs, or each haplotype block, our program will also provide additional parameters to partition each type of effects and its heritability, but it will cost much more time than prior choice.

Another highlight of our program is that it can be used to investigate the partitioned pairwise epistatic effects — that are divided into intra-chromosomes and inter-chromosomes. Any or all three types of pairwise epistatic effects could be partitioned, including additive-additive (AA), additive-dominant (AD), and dominant-dominant (DD) interactions. For each type, EPIHAP can estimate the variance components, heritability, as well as the genetic values and their reliability values for both intra- and inter-chromosomal epistatic effects.

2 Download

The latest version of EPIHAP, along with example data, can be downloaded from: https://github.com/AnimalGene/EPIHAP/tree/master. The precompiled 64-bit Linux executable of EPIHAP is compatible with any Linux distribution or Mac OS system.

3 Input Files

Before running EPIHAP, the user should prepare the following files as input:

- Parameter file
- SNP genotype file
- SNP map file
- Haplotype genotype file
- Phenotype file

3.1 Parameter file

EPIHAP requires a parameter file to read all necessary information. This file contains user-specific controls, output file names, and the full paths to other input files, such as the SNP genotype, haplotype genotype, and phenotype. The parameter file can be named anything. EPIHAP also can run without any input if a parameter file named "parameter.txt" exists in the current working directory. To execute this program, the user can choose either of the following two ways:

```
1) ./EPIHAP parameter.txt
2) ./EPIHAP
```

The lines starting with '#' sign are comments for the parameter definitions, which cannot be read by EPIHAP. Only the lines that start with a parameter name can be delivered to EPIHAP. The parameter name and its corresponding values in each non-comment line are delimited by a whitespace ' '. The capital letters 'Y' or 'N' observed in some non-comment lines are used to specify whether some certain parameters should be passed to EPIHAP or not (Y=Yes, N=No). EPIHAP will not work if the user deletes any line that does not start with the '#' sign in this file. The program is sensitive to the order of the parameters, which must not be rearranged. The interpretations for these parameters (with parameter names in **bold** font) are as follows:

geno_snp <string>

The string specifies the full path to the filename of the SNP genotype file for all chromosomes. **This file is required**, and the details regarding its format are outlined in **Section 3.2**.

Example: geno_snp /home/path/to/example.dat

geno_map <string>

The string specifies the full path to the filename of the SNP map file. **This file is required.** The details of the file format are delineated in **Section 3.2**.

Example: geno_map /home/path/to/example.map

use_geno_hap <Y/N>

The capital letters 'Y' or 'N' are used to indicate whether the parameter **geno_hap**, described below, should be passed to EPIHAP.

Example: use_geno_hap Y

geno_hap <string>

The string specifies the full path to the filename of the haplotype genotype file if the parameter **use_geno_hap** is set to 'Y'. The details of the file format are delineated in **Section 3.3**.

Example: geno_hap /home/path/to/example.hap

phenotype <string>

The string specifies the full path to the filename of the phenotype file. The details of the file format are described in **Section 3.4**.

Example: phenotype /home/path/to/example.phen

missing_phen_val <DOUBLE>

The double value is used to set the missing phenotype values [default=-9999]. This value must be same as the missing values that occurred in phenotype file.

Example: missing_phen_val -9999111

missing_hap_val <DOUBLE>

The double value is used to set the missing haplotype values [default=-9999]. This value must be same as the missing values that occurred in haplotype file.

Example: missing_hap_val -9999111

trait_col <INT>

The integer number defines the column of the desired trait of interest in the phenotype file.

Example: trait_col 7

factors_counts <INT>

The integer number is used to set the number of the fixed non-genetic factors only for discrete variables in the phenotype file. The parameter **factors_pos** will be skipped if the integer number is set to less than 1.

Example: factors_counts 2

factors_pos <INT> <INT> …

The integer numbers are used to set the positions of the fixed non-genetic factors only for discrete variables in the phenotype file if the parameter **factors_counts** ≥ 1.

Example: factors_pos 2 3

covar_counts <INT>

The integer number is used to set the number of the covariables in the phenotype file. The parameter **covar_pos** described below will be skipped if the integer number is set to less than 1.

Example: covar_counts 3

covar_pos <INT> <INT> …

The integer numbers are used to set the positions of the covariables in the phenotype file if the parameter **covar_counts** ≥ 1.

Example: covar_pos 4 5 6

make_grms <Y/N>

The ‘Y’ or ‘N’ letters are used to turn on/off running the construction of GRMs. If the parameter **load_grms** described below is set to ‘Y’, this parameter must be set to ‘N’.

Example: make_grms Y

make_partitioned_egrms <Y/N>

The ‘Y’ or ‘N’ letters are used to turn on/off running the construction of the GRMs for partitioned pairwise epistatic effects.

Example: make_partitioned_egrms Y

egrms_method <INT>

The integer numbers 1 or 2 are used to set which method is selected to construct epistatic GRMs [default=1]. 1: Approximate Genomic Epistasis Relationship Matrices (AGERM); 2: Exact Genomic Epistasis Relationship Matrices (EGERM).

Example: egrms_method 1

grm_prefix <string>

The string specifies the full path to the prefix filename of GRM files.

Example: grm_prefix /home/path/to/example

load_grms <Y/N>

The ‘Y’ or ‘N’ capital letters are used to turn on/off loading GRM files whose prefix filenames have been defined by the parameter **grm_prefix**, executing variance components estimation using GREML method and calculating heritability estimates and genetic values. This parameter must be set to ‘N’ if the parameter **make_grms** is set to ‘Y’.

Example: load_grms N

var_snp_a <DOUBLE>

The positive double value is utilized to establish the starting value of the additive variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_a** ≤ 0) to skip this parameter.

Example: var_snp_a 3

var_snp_d <DOUBLE>

The positive double value is used for the starting value of the dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_d** ≤ 0) to skip this parameter.

Example: var_snp_d 3

var_snp_aa <DOUBLE>

The positive double value is used for the starting value of the additive × additive variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_aa** ≤ 0) to skip this parameter. Furthermore, the **var_snp_aa** parameter should be skipped by EPIHAP if the parameter **make_partitioned_egrms** is set to 'Y'.

Example: var_snp_aa 6

var_snp_aa-inter <DOUBLE>

The positive double value is used for the starting value of the additive × additive variance component only for inter-chromosomes. The user can set an arbitrary value less than or equal to 0 (**var_snp_aa-inter** ≤ 0) to skip this parameter. The parameter **var_snp_aa-inter** should be skipped if the parameter **make_partitioned_egrms** is set to 'N'.

Example: var_snp_aa-inter 0

var_snp_aa-intra <DOUBLE>

The positive double value is used to set the starting value of the additive × additive variance component only for intra-chromosomes. The user can set an arbitrary value less than or equal to 0 (**var_snp_aa-intra** ≤ 0) to skip this parameter. The parameter **var_snp_aa-intra** should be skipped if the parameter **make_partitioned_egrms** is set to 'N'.

Example: var_snp_aa-intra 0

var_snp_ad <DOUBLE>

The positive double value is used to set the starting value of the additive × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_ad** ≤ 0) to skip this parameter. Furthermore, the **var_snp_ad** parameter should be skipped by EPIHAP if the parameter **make_partitioned_egrms** is set to ‘Y’.

var_snp_ad-inter <DOUBLE>

The positive double value is used to set the starting value of the inter-chromosomal additive × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_ad-inter** ≤ 0) to skip this parameter. The parameter **var_snp_ad-inter** should be skipped if the parameter **make_partitioned_egrms** is set to ‘N’.

var_snp_ad-intra <DOUBLE>

The positive double value is used to set the starting value of the intra-chromosomal additive × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_ad-intra** ≤ 0) to skip this parameter. The parameter **var_snp_ad-intra** should be skipped if the parameter **make_partitioned_egrms** is set to ‘N’.

var_snp_dd <DOUBLE>

The positive double value is used to set the starting value of the dominance × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_dd** ≤ 0) to skip this parameter. Furthermore, the **var_snp_dd** parameter should be skipped by EPIHAP if the parameter **make_partitioned_egrms** is set to ‘Y’.

var_snp_dd-inter <DOUBLE>

The positive double value is used to set the starting value of the inter-chromosomal dominance × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_dd-inter** ≤ 0) to skip this parameter. The parameter **var_snp_dd-inter** should be skipped if the parameter **make_partitioned_egrms** is set to ‘N’.

var_snp_dd-intra <DOUBLE>

The positive double value is used to set the starting value of the intra-chromosomal dominance × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_dd-intra** ≤ 0) to skip this parameter. The parameter **var_snp_dd-intra** should be skipped if the parameter **make_partitioned_egrms** is set to ‘N’.

var_snp_aaa <DOUBLE>

The positive double value is used to set the starting value of the additive × additive × additive variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_aaa** ≤ 0) to skip this parameter.

Example: var_snp_aaa 9

var_snp_aad <DOUBLE>

The positive double value is used to set the starting value of the additive × additive × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_aad** ≤ 0) to skip this parameter.

Example: var_snp_aad 7

var_snp_add <DOUBLE>

The positive double value is used to set the starting value of the additive × dominance × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_add** ≤ 0) to skip this parameter.

Example: var_snp_add 5

var_snp_ddd <DOUBLE>

The positive double value is used to set the starting value of the dominance × dominance × dominance variance component. The user can set an arbitrary value less than or equal to 0 (**var_snp_ddd** ≤ 0) to skip this parameter.

Example: var_snp_ddd 3

var_hap_a <DOUBLE>

The positive double value is used to set the starting value of the haplotype additive variance component. The user can set an arbitrary value less than or equal to 0 (**var_hap_a** ≤ 0) to skip this parameter.

var_e <DOUBLE>

The positive double value is used to set the starting value of the residual variance.

Example: var_e 1

num_iter <INT>

The integer number is used to set the maximum number of iterations that are allowed in the GREML_CE method [default=1000]. It works only when the parameter **cin_var** described below is set to ‘N’.

Example: num_iter 1000

ai-reml-iter-start <INT>

The integer number is used to set the starting iteration number for converting the EM-REML to the AI-REML to estimate the variance components [default=3].

Example: ai-reml-iter-start 3

tolerance <DOUBLE>

The positive double value is used to set the tolerance threshold as a convergence criterion to stop estimating the variance components [default=1E-8].

Example: tolerance 1.0E-08

tolerance_her <DOUBLE>

The positive double value is used to set the tolerance threshold as a convergence criterion to stop computing heritability estimates [default=1E-6].

Example: tolerance_her 1.0E-06

reml-ce-rel <Y/N>

The ‘Y’ or ‘N’ capital letters are used to turn on/off the calculation of the reliability of genetic values.

Example: reml-ce-rel N

marker_effects <Y/N>

The ‘Y’ or ‘N’ capital letters are used to turn on/off calculation of the genetic effects and heritability estimates partitioned by SNPs and/or haplotype blocks.

Example: marker_effects N

pairwise_effects <Y/N>

The ‘Y’ or ‘N’ capital letters are used to turn on/off computing the pairwise epistatic effects and heritability estimates that are partitioned by SNP pairs and printing the values to a specified file with the prefix of the filename defined by the parameter **output_gblup_prefix**. Furthermore, if this parameter is set to ‘N’, EPIHAP will ignore the parameter **num_pairwise_out**.

Example: pairwise_effects N

num_pairwise_out <INT>

The integer number is used to set the number of top-ranked SNP pairs with highest pairwise epistatic effects and heritability estimates [default = 30] if the parameter **pairwise_effects** is set to 'Y'.

Example: num_pairwise_out 30

cin_var <Y/N>

The 'Y' or 'N' letters are used to enable or disable the calculation of genetic values with the specified variance components. This parameter must be set to 'N' if the user decides to use GREML iterative method to estimate the variance components.

Example: cin_var N

output_gblup_prefix <string>

The string specifies the full path to the prefix filenames of main GBLUP output files. These files contain the estimated fixed non-genetic effects file, the GREML output file, the genetic values file, and the marker effects for all kinds of effect files if both the **marker_effects** and **pairwise_effects** parameters are set to 'Y'. The details of these files are described in **Section 4.1 – 4.6**.

Example: output_gblup_prefix /home/path/to/example

numThreads <INT>

The integer number is used to set the number of threads for parallel computing [default=16].

Example: numThreads 16

log_prefix <string>

The string is used to specify the full path to the prefix filename of log file. EPIHAP can create two log files. One is named <prefix of filename>`_for_make_grms.log` when the second column of the **make_grms** parameter is set to 'Y', the other named <prefix of filename>`_for_gblup.log` when the second column of the **load_grms** parameter is set to 'Y'. The details of these files are described in **Section 4.7**.

Example: log_prefix /home/path/to/example

An example of a parameter file for EPIHAP, as shown in **Box 1**, can be printed on the screen by running the following command:

`./EPIHAP -h` or`./EPIHAP --help`

Box 1: Example of a parameter file for EPIHAP

```
# The lines starting with '#' sign are comments for the parameter definitions, which
cannot be read by EPIHAP. Only the lines starting with the parameter name can be
delivered to EPIHAP. The parameter name and parameter values in each non-comment line
are delimited by a whitespace ' '.
# The capital letters 'Y' or 'N' observed in some non-comment lines are used to specify
whether some certain parameters should be passed to EPIHAP or not (Y=Yes, N=No).
# ####################################################################################
# Specify the full path to the filename of the genotype file.
geno_snp /home/path/to/example.dat
# Specify the full path to the filename of the SNP map file.
geno_map /home/path/to/example.map
# Set Y/N to specify whether the parameter geno_hap should be passed to EPIHAP or
not.
use_geno_hap Y
# Specify the full path to the filename of the haplotype genotypes file.
geno_hap /home/path/to/example.hap
# Specify the full path to the filename of the phenotype file.
phenotype /home/path/to/example.phen
# Set the missing phenotype values [default=-9999]. This value must be same as the
missing values occurred in phenotype file.
missing_phen_val -9999111
# Set the missing haplotype values [default=-9999]. This value must be same as the
missing values occurred in haplotype file.
missing_hap_val -9999111
# Set the position of the desired trait of interest in the phenotype file.
trait_col 7
# Set the number of the fixed non-genetic factors only for discrete variables in the
phenotype file. The parameter factors_pos will be skipped if the integer number is
set to less than 1.
factors_counts 2
# Set the positions of the fixed non-genetic factors only for discrete variables in
the phenotype file if the parameter factors_counts ⩾ 1.
factors_pos 2 3
# Set the number of the covariables in the phenotype file. The parameter covar_pos
will be skipped if the integer number is set to less than 1.
covar_counts 2
# Set the positions of the covariables in the phenotype file.
covar_pos 4 5 6
# Set Y/N to turn on/off running the construction of GRMs.
```

```
make_grms Y
# Set Y/N to turn on/off running the construction of GRMs for the pairwise epistatic effects (AA, AD and DD) that are partitioned into intra-chromosomes and inter-chromosomes.
make_partitioned_egrms N
# Set the method to construct epistatic GRMs via integers 1 or 2. 1: Approximate Genomic Epistasis Relationship Matrices (AGERM); 2: Exact Genomic Epistasis Relationship Matrices (EGERM), [default=1].
egrms_method 1
# Specify the full path to the prefix of GRM file names.
grm_prefix /home/path/to/grmfiles/example
# Set Y/N to turn on/off loading GRMs and executing variance components estimation using GREML method and calculating heritability estimates and genetic values.
load_grms N
# Set the starting value of the additive variance component. The user can set a starting value less than or equal to 0 (var_snp_a ⩽ 0) to skip this parameter.
var_snp_a 3
# Set the starting value of the dominance variance component.
var_snp_d 1
# Set the starting value of the additive × additive variance component.
var_snp_aa 6
# Set the starting value of the additive × additive variance component only for inter-chromosomes.
var_snp_aa-inter 0
# Set the starting value of the additive × additive variance component only for intra-chromosomes.
var_snp_aa-intra 0
# Set the starting value of the additive × dominance variance component.
var_snp_ad 4
# Set the starting value of the additive × dominance variance component only for inter-chromosomes.
var_snp_ad-inter 0
# Set the starting value of the additive × dominance variance component only for intra-chromosomes.
var_snp_ad-intra 0
# Set the starting value of the dominance × dominance variance component.
var_snp_dd 2
# Set the starting value of the dominance × dominance variance component only for inter-chromosomes.
var_snp_dd-inter 0
# Set the starting value of the dominance × dominance variance component only for intra-chromosomes.
```

```
var_snp_dd-intra 0
# Set the starting value of the additive × additive × additive variance component.
var_snp_aaa 9
# Set the starting value of the additive × additive × dominance variance component.
var_snp_aad 7
# Set the starting value of the additive × dominance × dominance variance component.
var_snp_add 5
# Set the starting value of the dominance × dominance × dominance variance component.
var_snp_ddd 3
# Set the starting value of the haplotype additive variance component.
var_hap_a 3
# Set the starting value of the residual variance.
var_e 1
# Set the maximum number of iterations that are allowed in the GREML_CE method
[default=1000].
num_iter 1000
# Set the starting iteration number for converting the EM-REML to the AI-REML to
estimate the variance components [default=3].
ai-reml-iter-start 3
# Set the tolerance threshold as a convergence criterion to stop estimating the
variance components [default=1E-8].
tolerance 1.0E-08
# Set the tolerance threshold as a convergence criterion to stop estimating the
heritability [default=1E-6].
tolerance_her 1.0E-06
# Set Y/N to turn on/off calculation of the reliability of genomic breeding values.
reml-ce-rel N
# Set Y/N to turn on/off computing the genetic effects and heritability estimates
partitioned by SNPs or haplotype blocks.
marker_effects N
# Set Y/N to turn on/off computing the pairwise epistatic effects and heritability
estimates that are partitioned by SNP pairs.
pairwise_effects N
# Set the number of top-ranked SNP pairs with highest pairwise epistatic effects and
heritability estimates [default = 30].
num_pairwise_out 30
# Set Y/N to turn on/off calculating the genetic values with the specified variance
components.
cin_var N
# Specify the full path to the prefix filenames of main GBLUP output files.
output_gblup_prefix /home/path/to/out/example
# Set the number of threads for parallel computing [default=16].
```

```
numThreads 16
# Set the full path to the prefix filename of log file.
log_prefix /home/path/to/log/example
```

3.2 Genotype Files

EPIHAP recognizes SNP genotypes file by the parameter **geno_snp**. This file is formatted as the genotype plain text file. This file organizes the data with rows representing individuals and columns representing SNP genotypes. The columns are tab or whitespaces delimited. The header line should have to start with the word `ID` in the very initial part of this file, while the following columns in this line should be the SNP IDs defined by the user. The individual IDs should exist in the first column.

By default, the SNP genotypes in this file are assumed to be encoded as 0=A1A1, 1=A1A2, and 2=A2A2, where “0” and “2” denote the two homozygous genotypes, and “1” denotes the heterozygous genotype. The missing genotypes are assumed to be encoded by other integers and will be treated as zero in GRMs construction. The general format of this file (**Box 2**) is as follows:

Box 2: Example of a SNP genotype plain text file

```
ID M1 M2 M3 M4 M5
Ind_1 0 0 2 0 2
Ind_2 1 0 2 0 2
Ind_3 0 0 1 0 2
```

Additionally, even though the SNP IDs have been given in the first line of the genotype plain text file, EPIHAP requires an SNP map file, which can be read by setting the parameter **geno_map** to provide more details about those SNPs. The map file whose columns are delimited by tabs or whitespaces has one row for each SNP. Each row has three columns: the first column is the chromosome number where the SNP is located, the second column is the SNP ID, and the last column is the SNP physical base pair position on the chromosome. The header row must include the keywords `Chr`, `SNPID`, and `Position`. The format of this file is as follows (**Box 3**):

Box 3: Example of a SNP map file

```
Chr    SNPID  Position
1      M1     2353975
1      M2     4283675
2      M3     2519921
3      M4     2829852
3      M5     5679578
```

Note that the SNP IDs in both the map file and the genotype plain text file should be arranged in the same order, and those SNPs must be sorted by chromosome number and then by physical position from small to large.

3.3 Haplotype File

EPIHAP can read the haplotype genotype file by the parameter **geno_hap**. The file format used in GVCHAP can be recognized when all chromosome files are merged into single. This file must be organized in a way that the data with rows representing individuals and columns representing haplotype genotypes. The file must be tab or whitespace delimited. The header line should begin with the word `ID` followed by the haplotype IDs defined by the user. The individual IDs should exist in the first column.

Following the description in the GVCHAP manual (Prakapenka et al., 2020), a haplotype genotype treated as an "allele" takes every two columns per row (Da, 2015). The general format of this file (**Box 4**) is as follows:

Box 4: Example of a haplotype genotype file

```
ID hap_2_1 hap_2_1 hap_2_2 hap_2_2 hap_2_3 hap_2_3
Ind_1 1 1 1 1 1 2
Ind_2 2 3 2 2 1 1
Ind_3 1 1 1 1 1 1
```

Note that the individual IDs in this file should be in the same order as those in the SNP genotypes file.

3.4 Phenotype File

EPIHAP can recognize the phenotype file by the parameter **phenotype**. It contains fixed non-genetic effects as well as numeric phenotypic values for one or more quantitative traits. The file is a space/tab-delimited and contains a header. The header row must begin with the word ID, followed by the column names specified by the user. Each row describes a single individual, with individual IDs located in the first column.

Both discrete and continuous variables can be treated as fixed non-genetic effects. Discrete variables typically include factors such as gender, herd, year, season, treatment, or living conditions. In contrast, continuous variables often encompass measures such as body weight or age. Here, we refer to covariables as the continuous variables that are also treated as fixed non-genetic effects. The general format of this file is as follows:

Box 5: Example of a phenotype file

```
ID     Fix1   Fix2   Cov1   Cov2   Cov3   Trait1 Trait2
Ind_1  2      0      36     93     21.1   1.75   5.96
Ind_2  1      0      40     102    29.6   1.61   -9999111
Ind_3  1      0      41     102    26.9   1.71   6.34
Ind_4  1      0      51     115    27.4   1.73   5.92
Ind_5  1      0      31     87     32.4   1.69   -9999111
Ind_6  1      0      56     92     26.1   1.64   6.30
```

Note that the sample size in this file must be equal to that in both the genotypic data file and the haplotype genotypes file. Consequently, if a phenotype for an individual is not recorded, the user should assign a double value as a missing value. EPIHAP will omit the missing values in each row after the parameter **missing_phen_val** is set in the parameter file (**BOX 1**). Not also that the individual IDs in this file should be in the same order as those in the SNP genotypes file and the haplotype genotypes file.

Furthermore, the user also can use the parameters **factors_pos** and **covar_pos** to specify the columns containing the fixed non-genetic effects (for discrete variables only) and the columns containing covariables, respectively (**BOX 1**).

4 Output Files

A total of ten output files can be generated by EPIHAP, including the GREML file, GBLUP file, SNP effects and heritability estimates file, three pairwise epistatic effect files for AA, AD and DD, haplotype block heritability estimates file, fixed non-genetic effects file, and two log files (one is for making GRMs, the other for running GBLUP). Some of these outputs are optional and others not. **The GREML file** contains the estimated variance components and heritability estimates for a particular mixed model. **The GBLUP file** contains genetic values along with reliability for all individuals that are from training and validation datasets. **The SNP effects and heritability estimates file** contains the additive and/or dominance effects and heritability estimates for each SNP. **Each of the three pairwise epistatic effect files** contains the specified number of top-ranked SNP pairs with highest pairwise epistatic effects for AA, AD or DD. **The haplotype block heritability estimates file** contains the haplotype additive heritability estimates for each haplotype block. **The two log files** contain information related to the implementation of EPIHAP.

4.1 GREML file (*._greml.txt)

This file with extension `_greml.txt` contains two sections. The first section reports the parameter values for all iterations that are estimated by using the iterative methods EM-REML or AI-REML.

Box 6: The first section of a GREML output file for the model A + AA + AH (part 1/2)

```
Iteration VA           Tolerance_VA  VAA          Tolerance_VAA
1         1.029109e-03 2.998971e+00  1.187135e-03 2.998813e+00
2         9.600906e-04 6.901881e-05  1.257439e-03 7.030379e-05
3         4.036442e-04 5.564464e-04  2.370585e-03 1.113147e-03
.
.
.
10        5.436292e-04 5.798440e-08  2.111835e-03 1.711129e-07
```

```
SE        1.181986e-03              3.068501e-03
```

Box 6: The first section of a GREML output file for the model A + AA + AH (part 2/2)

```
Iteration VAH          Tolerance_VAH VE           Tolerance_VE
1         1.061432e-03 2.998939e+00  3.902137e-04 9.996098e-01
2         1.016906e-03 4.452668e-05  4.080960e-04 1.788226e-05
3         4.318229e-04 5.850827e-04  2.124889e-04 1.956071e-04
.
.
.
10        5.902319e-04 6.760930e-08  2.852786e-04 5.071183e-08
SE        1.424063e-03               3.195845e-03
```

These parameters include the variance components and tolerance values for all types of genetic effects specified within a mixed model, as well as the residual variance and its corresponding tolerance value. The first line in this section is a header containing the column names, followed by a line that lists these estimated parameter values for each iteration. The last line in this section contains the standard errors (SEs) for all variance components. **Box 6** above is an example of the first section of the GREML output file, with the model set to be A + AA + AH.

The second section of this file gives the heritability estimates and their standard errors (SEs) for all kinds of genetic effects, as well as the heritability in the broad sense and its SE. The following example (**Box 7**) is for the model A + AA + AH.

Box 7: The second section of a GREML output file for the model A + AA + AH

```
Additive heritability, SE            : 1.201732e-01, 9.101242e-02
Additive X Additive heritability,SE   : 4.422982e-01, 2.921357e-01
Additive Haplotype heritability, SE   : 2.434344e-02, 1.282699e-01
Heritability in the broad sense, SE   : 5.868148e-01, 2.527004e-01
```

Additionally, the GREML file format for the mixed models that contain partitioned pairwise epistatic effects are similar with the mixed models that do not have these effects, such as the GREML output for the model described in **BOX 6** and **BOX 7**. The following example is the GREML file for the model A + AA-intra + AA-inter (**Box 8**):

Box 8: Example of a GREML output file for the model A + AA-intra + AA-inter (part 1/2)

```
Iteration VA           Tolerance_VA  VAA-intra     Tolerance_VAA-intra
1         1.047246e-03  2.998953e+00  1.075830e-03  2.998924e+00
2         1.032505e-03  1.474122e-05  1.086802e-03  1.097206e-05
3         9.111873e-04  1.213180e-04  1.627747e-03  5.409445e-04
.
.
.
10        9.329523e-04  2.835410e-09  1.730604e-03  7.190436e-09
SE        8.503745e-04                2.267989e-03
```

Box 8: Example of a GREML output file for the model A + AA-intra + AA-inter (part 2/2)

```
Iteration VAA-inter     Tolerance_VAA-inter VE           Tolerance_VE
1         1.066841e-03  2.998933e+00        3.553969e-04 9.996446e-01
2         1.068826e-03  1.985788e-06        3.559099e-04 5.129528e-07
3         6.472981e-04  4.215283e-04        3.528117e-04 3.098114e-06
.
.
.
10        5.990757e-04  3.487865e-09        2.879189e-04 6.013392e-09
SE        3.435274e-03                      3.106307e-03

Additive heritability, SE                          : 2.627627e-01, 2.309443e-01
Additive X Additive intra-chr heritability, SE     : 4.874184e-01, 6.244278e-01
Additive X Additive inter-chr heritability, SE     : 1.687275e-01, 9.663039e-01
Heritability in the broad sense, SE                : 9.189087e-01, 8.769006e-01
```

4.2 GBLUP file (*._gblup.csv)

This file with extension `_gblup.csv` reports the genetic values and reliability in a comma-delimited format. The first column lists the individual IDs, followed by the columns give genetic values and their reliability values for each type of genetic effect. The antepenultimate and penultimate columns give the genetic values and their reliability values that are the sum of those values for each type of genetic effect, respectively. The last column lists the labels indicating which dataset the individuals have been assigned to, where the label **`T`** refers to the training dataset, and the label **`V`** refers to the validation dataset. Each row represents an individual. The first line is a header beginning with the term `ID` and ending with the word `Train./Valid`. The example below is for the model A + AA + AH (**Box 9**).

Box 9: Example of a GBLUP file for the model A + AA + AH (part 1/2)

ID	GBLUP_A	Reliability_A	GBLUP_AA	Reliability_AA
176	0.001488	0.183181	-0.0215562	0.565985
317	-0.01171	0.188804	-0.0489034	0.550212
519	0.00593	0.190651	-0.00666722	0.587858
.				
.				
.				

Box 9: Example of a GBLUP file for the model A + AA + AH (part 2/2)

ID	GBLUP_AH	Reliability_AH	GBLUP_G	Reliability_G	Train./Valid.
176	-0.00626406	0.199775	-0.0263319	0.888391	T
317	-0.0182701	0.207912	-0.0788798	0.881642	T
519	0.00348448	0.20082	0.00274724	0.901785	T
.					
.					
.					

In addition, the GBLUP file format for the mixed models that contain partitioned pairwise epistatic effects are also similar with the mixed models that do not have these effects, such as the GBLUP output for the model described in **BOX 9**. The following example is the GBLUP output file for the model A + AA-intra + AA-inter (**Box 10**):

Box 10: Example of a GBLUP file for the model A + AA-intra + AA-inter (part 1/2)

ID	GBLUP_A	Reliability_A	GBLUP_AA-intra	Reliability_AA-intra
176	-0.000439298	0.316587	-0.0180348	0.480422
317	-0.022095	0.314517	-0.0427888	0.468617
519	0.00776478	0.338689	-0.00357435	0.485868
.				
.				
.				

Box 10: Example of a GBLUP file for the model A + AA-intra + AA-inter (part 2/2)

ID	GBLUP_AA-inter	Reliability_AA-inter	GBLUP_G	Reliability_G	Train./Valid.
176	-0.00767382	0.163865	-0.02615	0.88633	T
317	-0.0148356	0.156004	-0.07972	0.881887	T
519	-0.00261852	0.175208	0.001572	0.8973	T
.					
.					
.					

4.3 SNP effects and heritability estimates (*._sig_snp_effect.snpe)

This file with extension `_sig_snp_effect.snpe` provides the additive and/or dominance effects and heritability estimates for each SNP. The example is as follows (**Box 11**):

Box 11: Example of an output file with SNP effects and heritability estimates (part 1/4)

```
Chr  SNP          Pos       Effect_A       m_effect_A     Effect_D
 1   rs12410822   5351373   1.286306e-03   3.858917e-01   1.899934e-03
 1   rs7547331   17764434   1.188440e-02   3.565320e+00   5.751421e-04
 1   rs2985327   28992968   1.020676e-02   3.062028e+00   1.611674e-03
 1   rs10889978  41396453  -4.632438e-03  -1.389731e+00  -2.741595e-03
 1   rs10489487  55327699  -5.828696e-04  -1.748609e-01  -2.209286e-04
```

Box 11: Example of an output file with SNP effects and heritability estimates (part 2/4)

```
 m_effect_D    Effect_A2     m_effect_A2    Effect_D2     m_effect_D2   h2_mrk_A
 5.699803e-01  1.286306e-03  3.858917e-01   1.899934e-03  5.699803e-01  3.013260e-05
 1.725426e-01  1.188440e-02  3.565320e+00   5.751421e-04  1.725426e-01  2.572188e-03
 4.835021e-01  1.020676e-02  3.062028e+00   1.611674e-03  4.835021e-01  1.897248e-03
-8.224785e-01  4.632438e-03  1.389731e+00   2.741595e-03  8.224785e-01  3.908117e-04
-6.627857e-02  5.828696e-04  1.748609e-01   2.209286e-04  6.627857e-02  6.187158e-06
```

Box 11: Example of an output file with SNP effects and heritability estimates (part 3/4)

```
m_h2_mrk_A   h2_mrk_D     m_h2_mrk_D   H2_mrk       h2_mrk_norm_A  m_h2_mrk_norm_A
9.039780e-03 1.950660e-04 5.851981e-02 2.251986e-04 -6.586196e-01  -1.975859e+02
7.716565e-01 1.787535e-05 5.362606e-03 2.590064e-03  2.341272e+00   7.023817e+02
5.691743e-01 1.403649e-04 4.210948e-02 2.037613e-03  1.544772e+00   4.634316e+02
1.172435e-01 4.061729e-04 1.218519e-01 7.969846e-04 -2.329805e-01  -6.989416e+01
1.856147e-03 2.637595e-06 7.912785e-04 8.824753e-06 -6.868778e-01  -2.060633e+02
```

Box 11: Example of an output file with SNP effects and heritability estimates (part 4/4)

```
h2_mrk_norm_D   m_h2_mrk_norm_D   H2_mrk_norm    m_H2_mrk_norm
-3.088979e-03   -9.266938e-01    -5.929424e-01   -1.778827e+02
-5.437195e-01   -1.631158e+02     1.915001e+00    5.745002e+02
-1.699887e-01   -5.099662e+01     1.329126e+00    3.987377e+02
 6.410240e-01    1.923072e+02     1.343748e-02    4.031243e+00
-5.902117e-01   -1.770635e+02    -8.224073e-01   -2.467222e+02
```

This is a right-aligned file whose columns are delimited by one or more whitespaces. The results are summarized in such a way that the first line contains the column names, and subsequent lines contain the partitioned genetic effects (additive or dominance effects) and heritability values, one row per SNP. Table 1 below is a brief explanation for these keywords in the header line.

Table 1: The description for the keywords in the header line of an output file containing SNP effects and heritability estimates.

Keywords	Explanation
“Chr”	chromosome number
“SNPID”	SNP’s ID
“Pos”	The bp position
“Effect_A” or “Effect_D”	The additive or dominance effect for each SNP
“m_effect_A” or “m_effect_D”	The “Effect_A” or “Effect_D” times the total number of markers
“abs_effect_A” or “abs_effect_D”	The absolute value of the “Effect_A” or “Effect_D”
“h2_mrk_A” or “h2_mrk_D”	The additive or dominance heritability for each SNP
“m_h2_mrk_A” or “m_h2_mrk_D”	The “h2_mrk_A” or “h2_mrk_D” times the total number of markers
“H2_mrk”	The broad sense heritability for each SNP
“h2_mrk_norm_A” or “h2_mrk_norm_D”	The normalized value of “h2_mrk_A” or “h2_mrk_D”
“m_h2_mrk_norm_A” or “m_h2_mrk_norm_D”	The normalized value of “m_h2_mrk_A” or “m_h2_mrk_D”
“H2_mrk_norm”	The normalized value of “H2_mrk”
“m_H2_mrk_norm”	The normalized value of “H2_mrk” times the total number of markers

4.4 Pairwise epistatic effect files

Based on the mixed model that has been specified by the user, EPIHAP will produce up to three files containing the specified number of the top ranked SNP pairs with highest pairwise epistatic effects and heritability estimates. Each file with an extension (`*._AA_epi_effect.snpe`, `*._AD_epi_effect.snpe` or `*._DD_epi_effect.snpe`) corresponds to one of three types of epistatic effects (AA, AD and DD). Below is an example for the file with extension `_AA_epi_effect.snpe` (**Box 12**). This is a right-aligned file whose columns are delimited by one or more whitespaces.

Box 12: Example of an output file containing pairwise epistatic effects and heritability estimates for AA (part 1/2)

```
Chr1 SNP1       Pos1      Chr2 SNP2       Pos2       Effect_AA     m_effect_AA
 1   rs605060   85951042  4    rs2296040  37020071   2.607297e-03  1.169373e+02
 1   rs605060   85951042  6    rs10948947 55650761  -2.087325e-03 -9.361655e+01
 1   rs12040129 216995294 4    rs2296040  37020071  -2.135229e-03 -9.576503e+01
 2   rs17728164 609326    11   rs948851   57187617   2.252696e-03  1.010334e+02
 2   rs17728164 609326    11   rs7108536  131420147 -2.283852e-03 -1.024308e+02
```

Box 12: Example of an output file containing pairwise epistatic effects and heritability estimates for AA (part 2/2)

```
Effect_abs_AA m_effect_abs_AA h2_mrk_AA    m_h2_mrk_AA  h2_mrk_norm_AA m_h2_mrk_norm_AA
2.607297e-03  1.169373e+02    1.138259e-04 5.105090e+00 1.600370e+01   7.177658e+05
2.087325e-03  9.361655e+01    7.295253e-05 3.271921e+00 1.005122e+01   4.507973e+05
2.135229e-03  9.576503e+01    7.633946e-05 3.423825e+00 1.054447e+01   4.729194e+05
2.252696e-03  1.010334e+02    8.496994e-05 3.810902e+00 1.180134e+01   5.292903e+05
2.283852e-03  1.024308e+02    8.733656e-05 3.917045e+00 1.214600e+01   5.447481e+05
```

In each file the results are summarized in such a way that the first line contains the column names, and subsequent lines contain the partitioned genetic effects (additive or dominance effects) and heritability values, one row per pair of SNPs. Table **2** below is a brief explanation for those keywords in the header line.

Table 2: The description for the keywords in the header line of an output file containing the partitioned epistatic effects and heritability estimates.

Keywords	Explanation
“Chr1”	chromosome number for the first SNP

“SNP1”	First SNP’s ID
“Pos1”	The bp position for the first SNP
“Chr2”	chromosome number for the second SNP
“SNP2”	Second SNP’s ID
“Pos2”	The bp position for the second SNP
“Effect_AA”, “Effect_AD”, “Effect_DA” or “Effect_DD”	The AA, AD, DA or DD effects for each pair of SNPs
“m_effect_AA”, “m_effect_AD”, “m_effect_DA” or “m_effect_DD”	The “Effect_AA”, “Effect_AD”, “Effect_DA” or “Effect_DD” times the total number of SNP pairs.
“abs_effect_AA”, “abs_effect_AD”, “abs_effect_DA”, or “abs_effect_DD”	The absolute value of the “Effect_AA”, “Effect_AD”, “Effect_DA” or “Effect_DD”
“m_abs_effect_AA”, “m_abs_effect_AD”, “m_abs_effect_DA”, or “m_abs_effect_DD”	The absolute value of the “Effect_AA”, “Effect_AD”, “Effect_DA” or “Effect_DD” times the total number of SNP pairs.
“h2_mrk_AA”, “h2_mrk_AD”, “h2_mrk_DA” or “h2_mrk_DD”	The AA, AD, DA, or DD heritability for each pair of SNPs
“m_h2_mrk_AA”, “m_h2_mrk_AD”, “m_h2_mrk_DA” or “m_h2_mrk_DD”	The “h2_mrk_AA”, “h2_mrk_AD”, “h2_mrk_DA” or “h2_mrk_DD” times the total number of SNP pairs
“h2_mrk_norm_AA”, “h2_mrk_norm_AD”, “h2_mrk_norm_DA” or “h2_mrk_norm_DD”	The normalized value of “h2_mrk_AA”, “h2_mrk_AD”, “h2_mrk_DA” or “h2_mrk_DD”
“m_h2_mrk_norm_AA”, “m_h2_mrk_norm_AD”, “m_h2_mrk_norm_DA” or “m_h2_mrk_norm_DD”	The normalized value of “m_h2_mrk_AA”, “m_h2_mrk_AD”, “m_h2_mrk_DA” or “m_h2_mrk_DD”

Note that the values for the dominance × additive (DA) effects also will be included in the file with extension `_AD_epi_effect.snpe`.

4.5 Haplotype block heritability estimates (*_hap_effect.snpe)

This file with extension `_hap_effect.snpe` provides the heritability estimates for haplotype blocks. This is a right-aligned file whose columns are delimited by one or more whitespaces. This file has three columns per row, where the first column is the index of haplotype block starting from zero, the second column is the haplotype additive heritability for each block, and the last column is the standardized value of the haplotype additive heritability in the second column. The first line is a header with three keywords: `HAPID`, `h2_hap_AH` and `h2_hap_std_AH`. Below is an example of an output file for the partitioned haplotype heritability by blocks (**BOX 13**):

Box 13: Example of output file for haplotype heritability

```
HAPID      h2_hap_AH   h2_hap_std_AH
    0   9.104952e-04   -5.379154e-03
    1   4.004936e-04   -7.299725e-01
    2   1.672690e-04   -1.061330e+00
    3   1.774456e-04   -1.046872e+00
```

4.6 Fixed non-genetic effects (*._fixed_effect.txt)

This file with extension `_fixed_effect.txt` contains estimated fixed non-genetic effects by the best linear unbiased estimation (BLUE) at the end of the GREML iterations (**Box 14**).

Box 14: Example of estimates of fixed non-genetic effects

```
Fixed effect        Level name     Level          Value
           0                mu         1   1.165615e+01
           1                 1         0   6.086162e+00
           1                 2         1   5.569993e+00
           2                 0         0  -4.608519e-01
           2                 1         1   8.125373e-01
           3                 1         0  -2.209107e-05
           3                 0         1  -8.386171e-02
           3                 2         2  -8.994692e-02
           4        Covariable         1   2.137628e-02
           5        Covariable         1   2.550369e-03
           6        Covariable         1   7.875365e-02
           7        Covariable         1   1.001592e-02
           8        Covariable         1  -7.397074e-03
```

4.7 Log files

EPIHAP will generate two log files with extension `_for_make_grms.log` and `_for_gblup.log`. If the **make_grms** and **load_grms** parameters described in section 3.1 are set to ‘Y’ and ‘N’, respectively, EPIHAP will produce the log file with extension `_for_make_grms.log` for creating GRMs. Conversely, if the parameter **load_grms** is set to ‘Y’ and the parameter **make_grms** is set to ‘N’, EPIHAP will produce the log file with extension `_for_gblup.log` for GBLUP estimation.

This log file will document details of the run of the program, including the variance components and heritability estimates for each type of genetic effect, the time cost for each iteration, the number of SNPs or haplotype blocks, the number of individuals in the training and validation datasets, and the number of individuals in the genotypes, etc. Below is an example of the log file when EPIHAP is used for GRMs inference (**Box 15**):

Box 15: Example of a log file when EPIHAP is used for creating GRMs.

```
***********************EPIHAP log file****************************


Version                                            :1.0.0
The local date and time is                         :Mon Apr  1 00:41:10 2024


reading parameter file run_making_grm.dat ...


geno_snp example.map
geno_map example.map
use_geno_hap Y
geno_hap example.hap
phenotype example.phen
missing_phen_val -9999111
missing_hap_val -9999111
trait_col 7
```

```
factors_counts 2
factors_pos 2 3
covar_counts 3
covar_pos 4 5 6
make_grms Y
make_partitioned_egrms Y
egrms_method 1
grm_prefix ./grmfiles/example
load_grms N
var_snp_a 3
var_snp_d 1
var_snp_aa 6
var_snp_aa-inter 0
var_snp_aa-intra 0
var_snp_ad 4
var_snp_ad-inter 0
var_snp_ad-intra 0
var_snp_dd 2
var_snp_dd-inter 0
var_snp_dd-intra 0
var_snp_aaa 9
var_snp_aad 7
var_snp_add 5
var_snp_ddd 3
var_hap_a 3
var_e 1
num_iter 1000
ai-reml-iter-start 3
tolerance 1e-08
tolerance_her 1e-06
reml-ce-rel N
marker_effects N
pairwise_effects N
num_pairwise_out 30
```

```
cin_var N
output_gblup_prefix ./out/example
numThreads 16
log_prefix ./log/example

The number of SNPs is                              :300
The time (seconds) cost for reading genotypes is   :0
The method used for epistatic GRMs inference is    :AGRM method
The mean of the diagonal elements for KA is        :1.009109e+02
The mean of the diagonal elements for KD is        :3.915760e+01
The mean of the diagonal elements for KAA-inter is :6.296310e+02
The mean of the diagonal elements for KAA-intra is :9.622297e+03
The mean of the diagonal elements for KAD-inter is :2.466329e+02
The mean of the diagonal elements for KAD-intra is :3.730474e+03
The mean of the diagonal elements for KDD-inter is :1.036757e+02
The mean of the diagonal elements for KDD-intra is :1.445810e+03
The time (seconds) cost for SNP GRMs inference is  :1

The number of haplotype blocks is                  :110
The mean of the diagonal elements for KAH is       :1.056761e+02
The time (seconds) cost for AH GRMs inference is   :0

The time (seconds) cost for all GRMs inference is  :1
```

Furthermore, if the program is used to estimate heritability and genetic values, the log file looks like this (**Box 16**):

Box 16: Example of a log file when EPIHAP is used for GBLUP estimation (model: A + AA)

```
***********************EPIHAP log file****************************


Version                                            :1.0.0
```

```
The local date and time is                         :Mon Apr  1 00:49:10 2024


reading parameter file run_loading_grm.txt ...


geno_snp example.map
geno_map example.map
use_geno_hap Y
geno_hap example.hap
phenotype example.phen
missing_phen_val -9999111
missing_hap_val -9999111
trait_col 7
factors_counts 2
factors_pos 2 3
covar_counts 3
covar_pos 4 5 6
make_grms N
make_partitioned_egrms N
egrms_method 1
grm_prefix ./grmfiles/example
load_grms Y
var_snp_a 3
var_snp_d 0
var_snp_aa 6
var_snp_aa-inter 0
var_snp_aa-intra 0
var_snp_ad 0
var_snp_ad-inter 0
var_snp_ad-intra 0
var_snp_dd 0
var_snp_dd-inter 0
var_snp_dd-intra 0
var_snp_aaa 0
var_snp_aad 0
```

```
var_snp_add 0
var_snp_ddd 0
var_hap_a 0
var_e 1
num_iter 5
ai-reml-iter-start 3
tolerance 1e-08
tolerance_her 1e-06
reml-ce-rel Y
marker_effects N
pairwise_effects N
num_pairwise_out 30
cin_var N
output_gblup_prefix ./out/example
numThreads 16
log_prefix ./log/example

Loading GRMs from files with prefix:   ./grmfiles/example

The number of levels for descrete variable 1 is    :3
The number of individuals in genotypes file is     :100
The number of individuals in training dataset is   :99
The number of individuals in validation dataset is :1

Iteration: 1
EM-REML
VA            = 1.275967e-03   Tolerance_VA            = 2.998724e+00
VAA           = 2.861046e-03   Tolerance_VAA           = 5.997139e+00
VE            = 4.612615e-04   Tolerance_VE            = 9.995387e-01
The time (seconds) cost for this iteration is      :0

Iteration: 2
EM-REML
VA            = 1.182742e-03   Tolerance_VA            = 9.322496e-05
```

```
VAA           = 2.940476e-03   Tolerance_VAA          = 7.942969e-05
VE            = 4.587407e-04   Tolerance_VE           = 2.520801e-06

h2_A          = 2.581302e-01   Tolerance_h2_A         = 1.935798e-02
h2_AA         = 6.417509e-01   Tolerance_h2_AA        = 1.955093e-02
H2            = 8.998811e-01   Tolerance_H2           = 1.929536e-04
The time (seconds) cost for this iteration is      : 0

Iteration: 3
EM-REML
VA            = 1.103010e-03   Tolerance_VA           = 7.973171e-05
VAA           = 3.011471e-03   Tolerance_VAA          = 7.099496e-05
VE            = 4.547744e-04   Tolerance_VE           = 3.966301e-06

h2_A          = 2.413982e-01   Tolerance_h2_A         = 1.673198e-02
h2_AA         = 6.590726e-01   Tolerance_h2_AA        = 1.732168e-02
H2            = 9.004708e-01   Tolerance_H2           = 5.896991e-04
The time (seconds) cost for this iteration is      : 0

Iteration: 4
EM-REML
VA            = 1.034297e-03   Tolerance_VA           = 6.871317e-05
VAA           = 3.075286e-03   Tolerance_VAA          = 6.381560e-05
VE            = 4.496748e-04   Tolerance_VE           = 5.099523e-06

h2_A          = 2.268564e-01   Tolerance_h2_A         = 1.454182e-02
h2_AA         = 6.745147e-01   Tolerance_h2_AA        = 1.544208e-02
H2            = 9.013710e-01   Tolerance_H2           = 9.002607e-04
The time (seconds) cost for this iteration is      : 0

Iteration: 5
EM-REML
VA            = 9.746521e-04   Tolerance_VA           = 5.964455e-05
VAA           = 3.132965e-03   Tolerance_VAA          = 5.767860e-05
```

```
VE            = 4.436914e-04   Tolerance_VE            = 5.983432e-06

h2_A          = 2.141477e-01   Tolerance_h2_A          = 1.270869e-02
h2_AA         = 6.883658e-01   Tolerance_h2_AA         = 1.385109e-02
H2            = 9.025134e-01   Tolerance_H2            = 1.142395e-03
The time (seconds) cost for this iteration is      : 0

Inverse of AI matrix:

   1.241606e-06   6.470345e-08  -1.066646e-06
   6.470345e-08   1.987723e-05  -1.981621e-05
  -1.066646e-06  -1.981621e-05   2.097765e-05

The time (seconds) cost for all iterations is      : 0

Additive heritability, SE                          : 2.141477e-01, 2.366489e-01
Additive X Additive heritability, SE               : 6.883658e-01, 9.806914e-01
Heritability in the broad sense, SE                : 9.025134e-01, 1.005996e+00

The time (seconds) cost for GBLUP estimation is    : 0

The time (seconds) cost for GREML_CE is            : 0
```

Simultaneously, the warning information and execution process will be printed on the screen or written in the “*.stderr” file when the program is implemented on a cluster.

5 Tutorial

EPIHAP reads the parameters from a parameter file follows the program name. It is best to place the EPIHAP and all necessary files for its execution in the same directory. If these files are located elsewhere, the path for these files must be provided. In the following, we provide some examples of how to execute this program.

5.1 Data set

The folder `example` contains eight files: `example_parameter_step1.txt`, `example_parameter_step2.txt`, `example_parameter_step2_for_A+D+AA-intra+AA-inter.txt`, `example_parameter_for_marker_effects.txt`, `example.dat`, `example.map`, `example.hap` and `example.phen`. The parameter file `example_parameter_step1.txt`, `example_parameter_step2.txt`, `example_parameter_step2_for_A+D+AA-intra+AA-inter.txt`, and `example_parameter_for_marker_effects.txt` contains all parameters that can be read by EPIHAP. The genotypic data is stored in the `example.dat` file with a header line followed by 100 lines corresponding to 100 individuals sharing 300 SNPs. The SNP information is in the `example.map` file with a header line followed by 300 lines corresponding to 300 SNPs. The haplotype genotypes are in the `example.hap` file with a header line followed by 100 lines corresponding to 100 individuals sharing 110 haplotype blocks (every two columns per block). The phenotypic data is in the `example.phen` file with a header line followed by 100 lines corresponding to 100 individuals.

5.2 GRMs Inference

To reduce computing time, the first step is to infer the GRMs. EPIHAP uses the second definition of Q matrix to calculate those matrices. To start this step, the parameters **make_grms** and **load_grms** should be set to ‘Y’ and ‘N’, respectively. If the **make_partitioned_egrms** parameter is set to ‘N’, EPIHAP will produce the GRMs for the effects A, D, AA, AD, DD, AAA, AAD, ADD, DDD, and AH. The renamed parameter file used for this task is as follows:

example_parameter_step1.txt

```
geno_snp example.dat
```

```
geno_map example.map
use_geno_hap Y
geno_hap example.hap
phenotype example.phen
missing_phen_val -9999111
missing_hap_val -9999111
trait_col 7
factors_counts 2
factors_pos 2 3
covar_counts 3
covar_pos 4 5 6
make_grms Y
make_partitioned_egrms N
egrms_method 1
grm_prefix ./grmfiles/example
load_grms N
var_snp_a 3
var_snp_d 1
var_snp_aa 6
var_snp_aa-inter 0
var_snp_aa-intra 0
var_snp_ad 4
var_snp_ad-inter 0
var_snp_ad-intra 0
var_snp_dd 2
var_snp_dd-inter 0
var_snp_dd-intra 0
var_snp_aaa 9
var_snp_aad 7
var_snp_add 5
var_snp_ddd 3
var_hap_a 3
var_e 1
```

```
num_iter 1000
ai-reml-iter-start 3
tolerance 1.0E-08
tolerance_her 1.0E-06
reml-ce-rel N
marker_effects N
pairwise_effects N
num_pairwise_out 30
cin_var N
output_gblup_prefix ./out/example
numThreads 16
log_prefix ./log/example
```

To do this, create the directory to store GRM and log files and run EPIHAP:

```
mkdir grmfiles log
./EPIHAP example_parameter_step1.txt
```

EPIHAP will produce 12 binary files and three plain text files. The brief description of those files are as follows (**Table 3**):

Table 3: The description for the output files after executing GRMs inference.

File	Type	Description
example.g.A	Binary	The A GRM
example.g.D	Binary	The D GRM
example.g.AA	Binary	The AA GRM
example.g.AD	Binary	The AD GRM
example.g.DD	Binary	The DD GRM
example.g.AAA	Binary	The AAA GRM
example.g.AAD	Binary	The AAD GRM
example.g.ADD	Binary	The ADD GRM
example.g.DDD	Binary	The DDD GRM
example.g.AH	Binary	The AH GRM

example.gdiag	Binary	The mean of the diagonal elements for WW' matrix
example.indgeno	Binary	The individual IDs
example.g.WAT.txt	Plain text	The W’ matrix for A
example.g.WDT.txt	Plain text	The W’ matrix for D
example_for_make_grms.log	Plain text	The log file for making GRMs

By default, the GRMs for epistatic effects are calculated using the AGERM method. If the user decides to use EGERM to compute GRMs for epistatic effects, please set the parameter **egrms_method** to 2. Additionally, EPIHAP also can produce the GRMs for partitioned pairwise epistatic effects when the parameters **make_grms** and **make_partitioned_egrms** in the *example_parameter_step1.txt* parameter file is set to ‘Y’, and the parameter **load_grms** is set to ‘N’. EPIHAP will produce up to 11 binary and three plain text files. A brief description of these files is as follows (**Table 4**):

Table 4: The description for the output files after executing GRMs inference for partitioned pairwise epistatic effects.

File	Type	Description
example.g.A	Binary	The A GRM
example.g.D	Binary	The D GRM
example.g.AA-inter	Binary	The AA-inter GRM
example.g.AA-intra	Binary	The AA-intra GRM
example.g.AD-inter	Binary	The AD-inter GRM
example.g.AD-intra	Binary	The AD-intra GRM
example.g.DD-inter	Binary	The DD-inter GRM
example.g.DD-intra	Binary	The DD-intra GRM
example.g.AH	Binary	The AH GRM
example.gdiag	Binary	The mean of the diagonal elements for each WW' matrix
example.indgeno	Binary	The individual IDs
example.g.WAT.txt	Plain text	The W’ matrix for A
example.g.WDT.txt	Plain text	The W’ matrix for D

example_divided_for_make_grms.log	Plain text	The log file for making GRMs

5.3 GBLUP and Reliability Estimation

After creating the GRMs, in this step we use EPIHAP to estimate the genetic values and reliability for a specified model. To start this step, the parameters **make_grms** and **load_grms** should be set to 'N' and 'Y', respectively. Next, we give an example to show how to conduct genetic values and reliability estimation for the model A+D+AA+AH. In this case, the parameter **make_partitioned_egrms** also should be set to 'N'. For this model, we should use the following parameter file:

example_parameter_step2.txt

```
geno_snp example.dat
geno_map example.map
use_geno_hap Y
geno_hap example.hap
phenotype example.phen
missing_phen_val -9999111
missing_hap_val -9999111
trait_col 7
factors_counts 2
factors_pos 2 3
covar_counts 3
covar_pos 4 5 6
make_grms N
make_partitioned_egrms N
egrms_method 1
grm_prefix ./grmfiles/example
load_grms Y
var_snp_a 3
var_snp_d 1
var_snp_aa 6
```

```
var_snp_aa-inter 0
var_snp_aa-intra 0
var_snp_ad 0
var_snp_ad-inter 0
var_snp_ad-intra 0
var_snp_dd 0
var_snp_dd-inter 0
var_snp_dd-intra 0
var_snp_aaa 0
var_snp_aad 0
var_snp_add 0
var_snp_ddd 0
var_hap_a 3
var_e 1
num_iter 1000
ai-reml-iter-start 3
tolerance 1.0E-08
tolerance_her 1.0E-06
reml-ce-rel Y
marker_effects N
pairwise_effects N
num_pairwise_out 30
cin_var N
output_gblup_prefix ./out/example
numThreads 16
log_prefix ./log/example
```

After implementing the following command:

```
mkdir out
./EPIHAP example_parameter_step2.txt
```

EPIHAP will load the GRMs saved in the folder named “grmfiles” and produce the following four files: “example_gblup.csv”, “example_greml.txt”, “example_for_gblup.log” and “example _fixed_effect.txt". These files will be saved into the folder named "out". The details of these files are provided in **Section 4**.

Note that EM-REML and AI-REML iteration algorithms will be used in this step. If AI-REML produced any negative estimate of variance components, the program will return to EM-REML automatically. The user can set the iteration number from which the AI-REML will be used after using EM-REML algorithm to minimize the chance of failure of AI-REML by the parameter **ai-reml-iter-start**.

Additionally, EPIHAP also can estimate the genetic values and reliability for three partitioned pairwise epistatic effects. Before proceeding with this task, the **make_partitioned_egrms** parameter should be set to 'Y', and the starting values of the **var_snp_aa**, **var_snp_ad**, and **var_snp_dd** parameters, which must be skipped by EPIHAP, are set to be less than or equal to 0. For example, if we decide to run the model A+D+AA-intra+AA-inter, we need to set the starting values of the **var_snp_a**, **var_snp_d**, **var_snp_aa-intra**, **var_snp_aa-inter** and **var_e** parameters to be positive double values, and set the starting values for all other variance component parameters to be less than or equal to 0. An example run for this model is as follows:

```
 ./EPIHAP example_parameter_step2_for_A+D+AA-intra+AA-inter.txt
```

5.4 Calculation of the Partitioned Genetic Effects and Heritability Estimates

EPIHAP also can estimate the genetic effects and heritability for a SNP, a pair of SNPs or a haplotype block after estimating the variance components using GREML method. We here give an example for the model A + D + AA + AD + DD + AH to show how to create these output files. To do this, the **marker_effects**, **pairwise_effects**, and **cin_var** parameters should be set to 'Y', and the estimated values for the variance components of A, D, AA, AD, DD, AH and residual variance should be used as the starting values of the **var_snp_a**, **var_snp_d**, **var_snp_aa**, **var_snp_ad**, **var_snp_dd** , **var_snp_ah**, and **var_e** parameters, respectively.

Therefore, we should use the following parameter file:

example_parameter_for_marker_effects.txt

```
geno_snp example.dat
geno_map example.map
use_geno_hap Y
geno_hap example.hap
phenotype example.phen
missing_phen_val -9999111
missing_hap_val -9999111
trait_col 7
factors_counts 2
factors_pos 2 3
covar_counts 3
covar_pos 4 5 6
make_grms N
make_partitioned_egrms N
egrms_method 1
grm_prefix ./grmfiles/example
load_grms Y
var_snp_a 3
var_snp_d 1
var_snp_aa 6
var_snp_aa-inter 0
var_snp_aa-intra 0
var_snp_ad 4
var_snp_ad-inter 0
var_snp_ad-intra 0
var_snp_dd 2
var_snp_dd-inter 0
var_snp_dd-intra 0
var_snp_aaa 9
var_snp_aad 7
var_snp_add 5
var_snp_ddd 3
```

```
var_hap_a 3
var_e 1
num_iter 1000
ai-reml-iter-start 3
tolerance 1.0E-08
tolerance_her 1.0E-06
reml-ce-rel Y
marker_effects Y
pairwise_effects Y
num_pairwise_out 30
cin_var Y
output_gblup_prefix ./marker_effects/example
numThreads 16
log_prefix ./log/example_marker_effects
```

After implementing the following commands:

```
mkdir marker_effects
./EPIHAP example_parameter_for_marker_effects.txt
```

EPIHAP will produce eight files under the folder `marker_effects:`

- `example_gblup.csv`
- `example_greml.txt`
- `example_fixed_effect.txt`
- `example_snp_effect.snpe`
- `example_AA_effect.snpe`
- `example_AD_effect.snpe`
- `example_DD_effect.snpe`
- `example_haplotype_effect.snpe`

The `example_snp_effect.snpe` and `example_haplotype_effect.snpe` files can be used directly as inputs that are read by the program SNPEVG2 to create Manhattan plots (Wang et al., 2012). The SNPEVG2 and its manual are available at http://animalgene.umn.edu/. Note that, in this step EPIHAP cannot calculate the partitioned pairwise epistatic effects for each SNP pair.

Author Contributions

YD conceived this study. ZL is the author of the EPIHAP program. ZL and DP provided extensive evaluation that improved EPIHAP program. ZL, YD and DP prepared the user manual.

Acknowledgements

This research was supported by USDA National Institute of Food and Agriculture Grant no. 2011-67015-30333 and 2018-67015-28128, and by project MN-16-124 of the Agricultural Experiment Station at the University of Minnesota. Supercomputer computing time was provided by the Minnesota Supercomputer Institute at the University of Minnesota.

References

Da, Y. (2015). Multi-allelic haplotype model based on genetic partition for genomic prediction and variance component estimation using SNP markers. *BMC Genet* 16**,** 144. doi: 10.1186/s12863-015-0301-1.

Da, Y., Liang, Z., and Prakapenka, D. (2022). Multifactorial methods integrating haplotype and epistasis effects for genomic estimation and prediction of quantitative traits. *Front Genet* 13**,** 922369. doi: 10.3389/fgene.2022.922369.

Henderson, C.R. (1985). Best Linear Unbiased Prediction of Nonadditive Genetic Merits in Noninbred Populations. *Journal of Animal Science* 60(1)**,** 111-117.

Jiang, Y., and Reif, J.C. (2020). Efficient Algorithms for Calculating Epistatic Genomic Relationship Matrices. *Genetics* 216(3)**,** 651-669. doi: 10.1534/genetics.120.303459.

Prakapenka, D., Wang, C., Liang, Z., Bian, C., Tan, C., and Da, Y. (2020). GVCHAP: A Computing Pipeline for Genomic Prediction and Variance Component Estimation Using Haplotypes and SNP Markers. *Front Genet* 11**,** 282. doi: 10.3389/fgene.2020.00282.

Wang, C., Prakapenka, D., Wang, S., Pulugurta, S., Runesha, H.B., and Da, Y. (2014). GVCBLUP: a computer package for genomic prediction and variance component estimation of additive and dominance effects. *BMC Bioinformatics* 15**,** 270. doi: 10.1186/1471-2105-15-270.

Wang, S., Dvorkin, D., and Da, Y. (2012). SNPEVG: a graphical tool for GWAS graphing with mouse clicks. *BMC Bioinformatics* 13**,** 319. doi: 10.1186/1471-2105-13-319.