MVBLUP-Multi-view-BLUP

Version 1.0

Updated at 2024.07.15

yangwenyu@mail.hzau.edu.cn

yjianbing@mail.hzau.edu.cn

hj_xiong@mail.hzau.edu.cn

wubingjie@webmail.hzau.edu.cn

1 INPUT

1.1 Multi_view file

The Multi_view file requires a header row, which shows the names of characteristics. The first column shows the names of individuals. For the genotype file, the genotypes (e.g., AA, AT, TT) need to be converted into integer format as 0, 1, or 2 based on allele frequency (Table 1). For transcriptome data, metabolome data, and other characteristic data, the content must also be in a numeric format (Table 2).

Table 1 Genotypic format of 5 characteristics in 5 individuals

ID	characteristic1	characteristic1	characteristic1	characteristic1	characteristic1
sample1	0	0	0	0	0
sample2	0	0	0	0	0
sample3	0	2	2	0	2
sample4	0	2	0	0	0
sample5	0	0	2	0	0

Table 2 Transcriptomic format of 5 characteristics in 5 individuals

ID	characteristic1	characteristic1	characteristic1	characteristic1	characteristic1
sample1	16.10370	15.62257	154.2729	21.48103	9.37354
sample2	28.68876	20.92862	165.0147	31.95128	11.27726
sample3	35.62518	16.28580	119.0899	18.32152	15.26793
sample4	28.68876	20.92862	165.0147	31.95128	11.27726
sample5	28.68876	20.92862	165.0147	31.98128	11.27726

1.2 Phenotypic file

The phenotypic file must contain a header row, which represents the names of traits. The first column shows the names of the individuals (Table 3). The trait values are numeric and missing values are indicated as NA, following the R coding requirement.

Table 3 Phenotypic format of 2 traits in 5 individuals

ID	trait1	trait2
sample1	79.52	73.17
sample2	81.44	92.27
sample3	70.17	78.06
sample4	79.81	83.03
sample5	65.53	41.33

2 USAGE

2.1 Before analysis

To run MVBLUP, the user needs to prepare phenotypic data and multi-view data files. This manual uses one genomic data and seven transcriptomic data as examples of multi-view data. The user must ensure all files maintain the same order of individuals prior to analysis.

2.2 Read files

The user could use following codes and demo data we provided to test the function of reading data from disk to R working space. It's noted that, for novice users, the full path of data files that located in the computer is recommended to avoid possible errors.

```
# read multi-view data
Multi_view1 <- read.table("./data/Multi_view1.txt",
                          header=TRUE,
                          sep="",
                          stringsAsFactors=0,
                          check.names=FALSE)

# read phenotype
Phenotype <- read.table("./data/Phenotype.txt",
                        header = TRUE,
                        sep = "",
                        stringsAsFactors = 0,
                        check.names = FALSE)
```

2.3 Divide a population into training and test set

To execute following analyses, the whole population requires to be divided into training and test sets. The user requires to prepare a text table for set information of individuals. It can be a two-column table, where the first column is the names of individuals (must be the same to the data files) and the second column is the set identity as training and test. The user could read the set information using following codes.

```
train_test_id <- read.table("./data/train_test_id.txt",
                            header = TRUE,
                            sep = "",
                            stringsAsFactors = 0,
                            check.names = FALSE)
```

2.4 Obtain the multi-view kinship matrix

We have defined a similarity function to quantify the similarity between two individuals and employed it to calculate the multi-view kinship matrix. This calculation process consists of two steps. Initially, we harness the inner product of single-view datasets, which forms the cornerstone of the initial relationship matrix for these datasets. For this purpose, we devised a function titled "*similarity_inner_product*". Subsequently, after obtaining the weights for each view dataset by the MVBLUP model, we perform a weighted summation of the initial single-view kinship matrices to obtain the multi-view initial kinship matrix. By incorporating this matrix into our similarity function, we derive a refined multi-view kinship matrix. To operationalize this methodology, we developed a function named "*similarity_kinship*". The tutorial code can be validated as

specified below:

```
A_N <- similarity_inner_product(Multi_dataset, n)
A <- similarity_kinship(A_M)
```

Notes:

1. *Multi_dataset: a list composed of multi-view datasets.*
2. *n: the number of multi-view datasets.*
3. *A_M: a multi-view initial kinship matrix, constructed through the weighted integration of inner products derived from single-view data.*

2.5 Optimizing the weights of single-view datasets

In the training set, a differential evolution algorithm was constructed to search for an optimal set of weights, which represent the importance of different single-view data in phenotype prediction. If n single-view data are used in the MVBLUP model for learning, the function " *MVBLUP_weights* " will obtain n optimal weight values. If the early stopping criterion is met at the lth iteration during the training process, the function will record the number of iterations jumped out, l. The tutorial code can be tested as followed:

```
MVBLUP_results <- MVBLUP_weights(NP, n,
                                 CR, Mu,
                                 s0, s1,
                                 thre, iter,
                                 cv, trid,
                                 A_N,
                                  Phenotype,
                                 train_test_id)
```

Notes:

1. *NP: the size of population.*
2. *n: the number of multi-view datasets.*
3. *CR: crossover factor.*
4. *Mu: scaling factor.*
5. *s0: the lower bounds of the search space for weights.*
6. *s1: the upper bounds of the search space for weights.*
7. *thre: threshold of the early stopping mechanism.*
8. *iter: maximum iteration.*
9. *cv: the number of folds for cross validation.*
10. *trid: which trait.*
11. *A_N: the single-view inner product for calculating multi-view kinship matrix.*
12. *Phenotype: demonstrative phenotypic data with 100 individuals and 2 traits.*
13. *train_test_id : file that identifies the sample IDs for the training and test sets.*

2.6 Testing accuracy of MVBLUP model

In the test set, the above n optimal weight values are integrated with multi-view data for prediction, and the correlation between the observed values and predicted values in the test set is calculated, which is defined as the accuracy of the MVBLUP model. The user can use the function '*MVBLUP_PreTest*' to estimate the accuracy of MVBLUP method in current data:

```
MVBLUP_accuracy <- MVBLUP_PreTest(kinship_test,
                                  testop1_ex,
                                  trainop1_ex)
```

Notes:

1. *kinship_test: multi-view kinship matrix obtained from the MVBLUP model.*
2. *testop1_ex: true phenotypes of the test set.*
3. *trainop1_ex: true phenotypes of the train set.*

3 OUTPUT

3.1 Output of weights of different single-view datasets

The function "*MVBLUP_weights*" can learn the optimal weights for different views of datasets to maximize genomic prediction performance. Users can obtain the optimal weights using "*MVBLUP_results$MVBLUP_information*", which is a data frame containing two columns. The first column includes the names and basic information of the parameters obtained from the MVBLUP model, while the second column lists the optimal weights for the corresponding single-view data, the number of iterations required for convergence, and the trait name (Table 4). The initial eight rows depict the optimal weights for the eight single-view datasets. The ninth row signifies the number of iterations needed for successful training of the MVBLUP model. Lastly, the final row specifies the phenotype being predicted.

Table 4 Weight values and basic parameters obtained from training the MVBLUP model on the training set

parameters	values
View1_W	0.2206
View2_W	0.1660
View3_W	0
View4_W	1
View5_W	0.0088
View6_W	0.0296
View7_W	0.8944
View8_W	0.6403
iter	46
Trait	trait1

3.2 Output the predicted phenotypes and accuracy for the test set

The function "*MVBLUP_PreTest*" calculates the accuracy of the MVBLUP model, and users can obtain the predicted values and prediction accuracy for the test set using "*MVBLUP_accuracy*" (Table 5).

Table 5 Predicted phenotypes and accuracy for the test set

name	value
sample1	71.3280
sample2	65.7345
sample3	88.3808
…	…
Accuracy	0.9459

3.3 Visualize the learning workflow of the MVBLUP algorithm

Users can visualize the learning workflow of the MVBLUP algorithm using the '*plot_learning*' function, which requires the support of ggplot2 and RColorBrewer packages. This function generates two graphs: 1) the iterative training process with eight different data types; 2) changes in optimal weights during the MVBLUP model's learning process (Fig. 1).

```
plot_learning(MVBLUP_results$Training_Accuracy,
              MVBLUP_results$Training_Weight,
              MVBLUP_results$MVBLUP_information,
              output_path)
```

Notes:

1. *MVBLUP_results$Training_Accuracy: the average accuracy obtained from each iteration of five-fold cross-validation on the training set.*
2. *MVBLUP_results$Training_Weight: the optimal weights obtained in each iteration.*
3. *MVBLUP_results$MVBLUP_information: the optimal weight values and the number of iterations required for convergence*
4. *output_path: the path for storing the output results.*

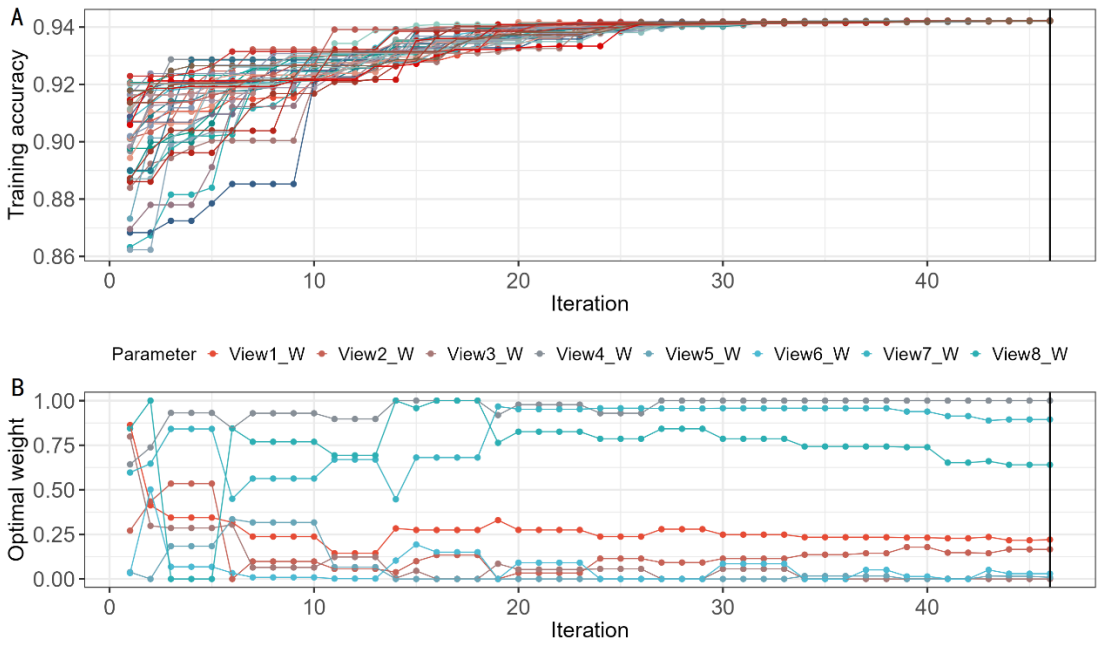

Fig. 1 The learning workflow of the MVBLUP algorithm