Tutorial of the MultiGS-R (v1.0)

Java Pipeline for Genomic Selection of Multiple Single Traits Using R-Based Models and Diverse Marker Types

This tutorial provides step-by-step instructions for running three use cases of **MultiGS-R**:

1. **Cross-validation (CV)**,
2. **Across-population prediction (APP) without test phenotypes**, and
3. **Across-population prediction (APP) with test phenotypes**.

It assumes that the pipeline and all required dependencies have been installed correctly.

Pipeline Setup and Example Data

1. Ensure R and all required packages are installed as listed in the README.

2. Clone the repository and enter the example folder:

```
> git clone https://github.com/AAFC-ORDC-Crop-Bioinfomatics/MultiGS-R.git
Cloning into 'MultiGS-R'...
remote: Enumerating objects: 269, done.
remote: Counting objects: 100% (269/269), done.
remote: Compressing objects: 100% (237/237), done.
remote: Total 269 (delta 98), reused 173 (delta 31), pack-reused 0 (from 0)
Receiving objects: 100% (269/269), 15.95 MiB | 12.19 MiB/s, done.
Resolving deltas: 100% (98/98), done.
> cd MultiGS-R/example/
>
```

3. Review the example genotype, phenotype, and configuration file.

Cross-Validation (CV)

1. Input data: example/inputFile/train_genotype.vcf and example/inputFile/train_phenotype.txt
2. Configure Cross-Validation

   Edit the configuration file (e.g., `MultiGS-R_config_CV.ini`) to define the **cross-validation parameters**. In this file, specify the **marker type** (SNP, HAP, or PC), **GS models**, **number of replicates**, **input genotype and phenotype files**, and the **output directory**, among other settings.

You can begin by using the parameters provided in **`MultiGS-R_config_CV.ini`**, **But you MUST update the rtm_gwas_snpldb_path and RScriptPath to point to your own tools.**

 # Haplotype block identification tool
rtm_gwas_snpldb_path = /path/to/rtm-gwas-snpldb
 # R path
RScriptPath = /path/to/Rscript

3. Run Cross-Validation

Run the pipeline script using the example configuration:

```
>
> java -Xmx20g -jar ../pipeline/MultiGS-R-1.0.jar  MultiGS-R_config_CV.ini

Check configuration...
✓ All configuration parameters validated successfully

[Log] Writing console output to: outputFile_CV_SNP/gs_69301007157951645.log

======================================================================
OmniGS-R (1.0): GENOMIC SELECTION PIPELINE USING R PACKAGES
======================================================================
Mode: CROSS-VALIDATION
Enabled models: [RR-BLUP, GBLUP, BRR, BL, BayesA, BayesB, BayesC, RFR, SVR, RKH
Feature (marker) view: SNP
Results directory: outputFile_CV_SNP
```

4. Results and output Figures

-- The output folder (outputFile_CV_SNP) contains detailed cross validation results (all_CV_results.txt) and summary statistics (CV_summary_statistics.csv) for all the replications and folds.

```
> ls -1 outputFile_CV_SNP/
all_CV_results.txt
CV_summary_statistics.csv
gs_69301215310516517.log
intermediate_data
pheno_data
plots
trait_predictions
>
```

(1) Pre-analysis of phenotypic data: Distribution of phenotypic values for the training population (pheno_data/YLD_hist.png)

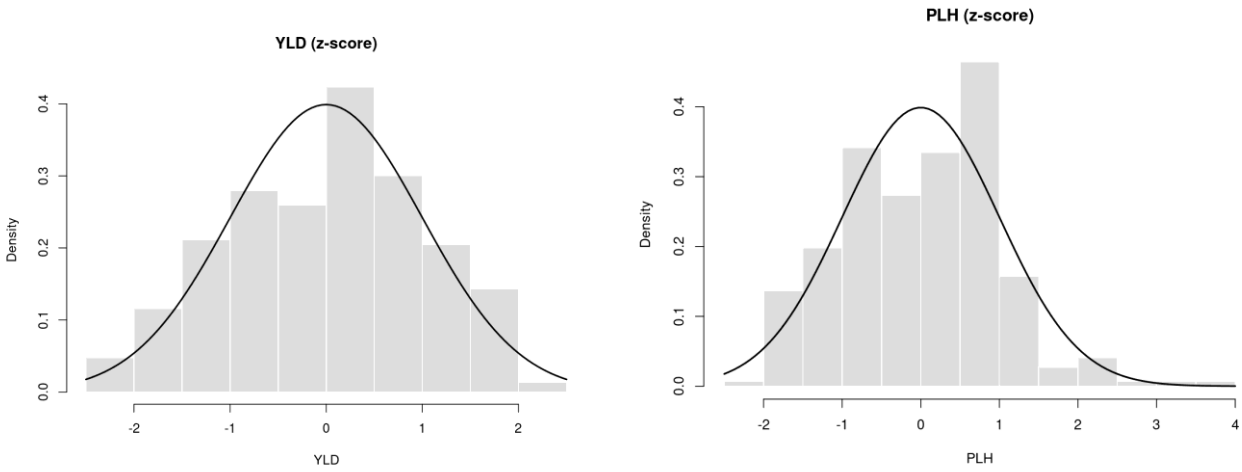

Figure 1: Histograms f the phenotypic values for YLD (A) and PLH (B).

(2) Summary of the cross-validation results across different models and traits (plots/SNP_model_trait_box_r_plot.png) (**Figure 2**)

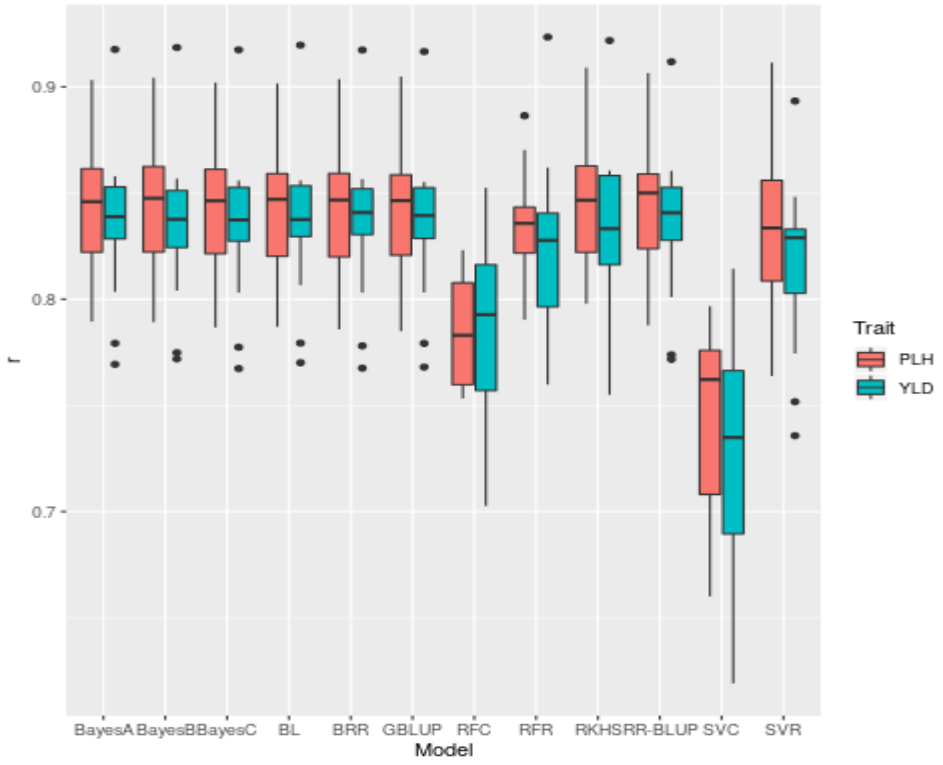

Figure 2. Boxplot of prediction accuracies across models and traits using a five-fold cross-validation scheme.

(3) Summary statistics of genomic selection (**Table 1**)

Table 1. Summary statistics of genomic selection using five-fold cross-validation scheme

	A	B	C	D	E	F	G	H
1	Trait	Model	mean_r	sd_r	mean_R2	sd_R2	count	
2	YLD	RR-BLUP	0.835	0.041	0.95	0.013	15	
3	YLD	GBLUP	0.833	0.041	0.94	0.01	15	
4	YLD	BRR	0.833	0.041	0.94	0.01	15	
5	YLD	BL	0.833	0.041	0.935	0.012	15	
6	YLD	BayesA	0.833	0.041	0.938	0.011	15	
7	YLD	BayesB	0.832	0.042	0.939	0.01	15	
8	YLD	BayesC	0.833	0.041	0.939	0.011	15	
9	YLD	RFR	0.822	0.039	0.97	0.002	15	
10	YLD	SVR	0.822	0.039	0.976	0.006	15	
11	YLD	RKHS	0.828	0.039	0.96	0.009	15	
12	YLD	RFC	0.78	0.037	0.727	0.009	15	
13	YLD	SVC	0.745	0.044	0.681	0.01	15	
14	PLH	RR-BLUP	0.844	0.032	0.956	0.006	15	
15	PLH	GBLUP	0.843	0.033	0.941	0.005	15	
16	PLH	BRR	0.843	0.034	0.941	0.005	15	
17	PLH	BL	0.844	0.033	0.934	0.006	15	
18	PLH	BayesA	0.844	0.033	0.938	0.005	15	
19	PLH	BayesB	0.844	0.032	0.94	0.004	15	
20	PLH	BayesC	0.844	0.033	0.94	0.005	15	
21	PLH	RFR	0.843	0.032	0.97	0.001	15	
22	PLH	SVR	0.835	0.035	0.985	0.002	15	
23	PLH	RKHS	0.852	0.033	0.964	0.004	15	
24	PLH	RFC	0.798	0.033	0.698	0.011	15	
25	PLH	SVC	0.752	0.049	0.655	0.011	15	

Prediction without test phenotype data

1. Input data:

   example/inputFile/train_genotype.vcf and example/inputFile/train_phenotype.txt

   example/inputFile/test_genotype.vcf

2. Configure prediction without test Phenotype data

   Edit the configuration file (e.g., `MultiGS-R_config_prediction1.ini`) to define the **across-population prediction (APP) parameters**. In this file, specify the **marker type** (SNP, HAP, or PC), **GS models**, **input genotype and phenotype files**, and the **output directory**, among other settings.

The main difference between **APP** and **cross-validation (CV)** is that **marker data for the test lines must be provided** in APP.

You can keep the same parameters as the file **`MultiGS-R_config_prediction1.ini`**. **But you MUST update the rtm_gwas_snpldb_path and RScriptPath to point to your own tools.**

 # Haplotype block identification tool
rtm_gwas_snpldb_path = /path/to/rtm-gwas-snpldb
 # R path
RScriptPath = /path/to/Rscript

3. Run the pipeline script using the example configuration:

```
>
> java -Xmx20g -jar ../pipeline/MultiGS-R-1.0.jar  MultiGS-R_config_prediction1.ini

Check configuration...
✓ All configuration parameters validated successfully

[Log] Writing console output to: outputFile_prediction_SNP_no_test_pheno/gs_69307543

================================================================
OmniGS-R (1.0): GENOMIC SELECTION PIPELINE USING R PACKAGES
================================================================
Mode: PREDICTION
```

4. Results and output Figures.

   --The output folder (outputFile_prediction_SNP_no_test_pheno) contains detailed prediction results (prediction_detailed_results.txt)

```
>
> ls -1 outputFile_prediction_SNP_no_test_pheno/
gs_69307543570303432.log
intermediate_data
pheno_data
plots
prediction_detailed_results.txt
trait_predictions
>
```

(1) Scatter plot of multidimensional scaling (MDS) dimensions of all training and test lines (Combined_mds.png) (**Figure 3**)

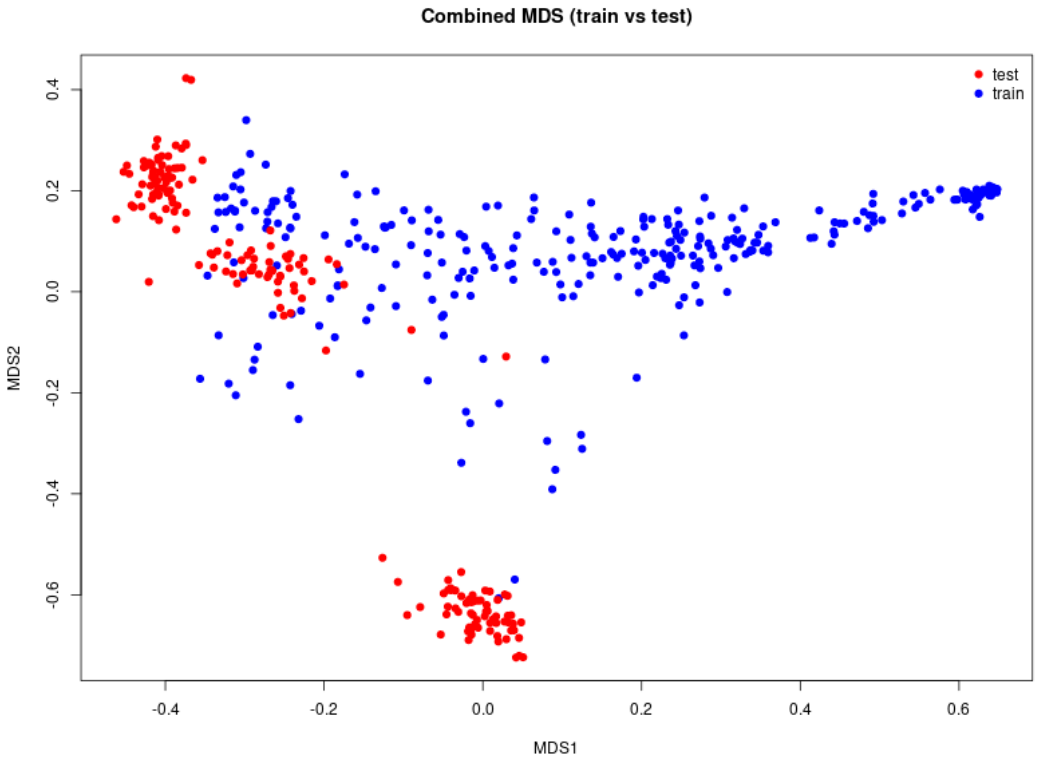

Figure 3. Scatter plot of the first two multidimensional scaling (MDS) dimensions illustrating genomic relatedness between training and test lines

(2) The distribution of the phenotypic values for training population (pheno_data/YLD_hist.png, pheno_data/PLH_hist.png, the same figure as in the cross-validation section)

(3) The GEBVs for test lines for YLD (**Table 2**) and PLH (**Table 3**) are stored in the folder **trait_predictions**

Table 2. Genomic estimated breeding values (GEBV) for YLD across models using SNP markers

	A	B	C
1	sample	predicted	
2	test43	1.069264	
3	test5	1.218637	
4	test14	1.432423	
5	test32	1.335674	
6	test33	1.729605	
7	test15	1.111455	
8	test23	0.734115	
9	test13	1.048203	
10	test12	0.856327	
11	test48	1.241334	
12	test51	0.744822	
13	test26	1.092827	
14	test4	1.202522	
15	test16	0.730979	
16	...	...	
17	...	...	

Table 3. Genomic estimated breeding values (GEBV) for PLH across models using SNP markers

	A	B	C
1	sample	predicted	
2	test43	0.697734	
3	test5	0.725268	
4	test14	0.815608	
5	test32	0.884929	
6	test33	1.031263	
7	test15	0.744502	
8	test23	0.423927	
9	test13	0.762388	
10	test12	0.817155	
11	test48	0.91558	
12	test51	0.43352	
13	test26	0.829277	
14	test4	1.00991	
15	test16	0.533918	
16	...	...	
17	...	...	

Prediction with test phenotype data

1. Input data:

   example/inputFile/train_genotype.vcf and example/inputFile/train_phenotype.txt

   example/inputFile/test_genotype.vcf and example/inputFile/test_phenotype.txt

2. Configure prediction with known test phenotype data

   Edit the configuration file (e.g., **`MultiGS-R_config_prediction2.ini`**) to define the **across-population prediction (APP) parameters**. In this file, specify the **marker type** (SNP, HAP, or PC), **GS models**, **input genotype and phenotype files**, and the **output directory**, among other settings.

   M**arker data (vcf) and phenotypic data for test lines are required in this case.**

   You can keep the same parameters as the file **`MultiGS-R_config_prediction2.ini`**. **But you MUST update the rtm_gwas_snpldb_path and RScriptPath to point to your own tools.**

   # Haplotype block identification tool
   rtm_gwas_snpldb_path = /path/to/rtm-gwas-snpldb
   # R path
   RScriptPath = /path/to/Rscript

3. Run the pipeline script using the example configuration:

```
>
> java -Xmx20g -jar ../pipeline/MultiGS-R-1.0.jar  MultiGS-R_config_prediction2.ini

Check configuration...
✓ All configuration parameters validated successfully

[Log] Writing console output to: outputFile_prediction_SNP_with_test_pheno/gs_693103011

================================================================
OmniGS-R (1.0): GENOMIC SELECTION PIPELINE USING R PACKAGES
================================================================
Mode: PREDICTION
```

4. Results and output figures

All the result files are stored in the folder **outputFile_prediction_no_test_pheno**

```
> ls -1 outputFile_prediction_SNP_with_test_pheno/
gs_692973708679679l3.log
intermediate_data
pheno_data
plots
prediction_detailed_results.txt
trait_predictions
>
```

(1) Scatter plot of multidimensional scaling (MDS) dimensions of all training and test lines (Combined_mds.png) (Same as before)
(2) The distribution of the phenotypic values for training population (pheno_data/YLD_hist.png, pheno_data/PLH_hist.png, same as before)
(3) The predict values for phenotype value for test lines are stored in the folder trait_predictions (same as before)

(4) **Scatter plots of observed phenotypic values versus GEBVs for the test population across all models,** when phenotypic data for all or some test lines are available. These plots can be used to assess the **prediction accuracy** of across-population prediction. **Figure 4** shows an example of the prediction results obtained using the **GBLUP** model.

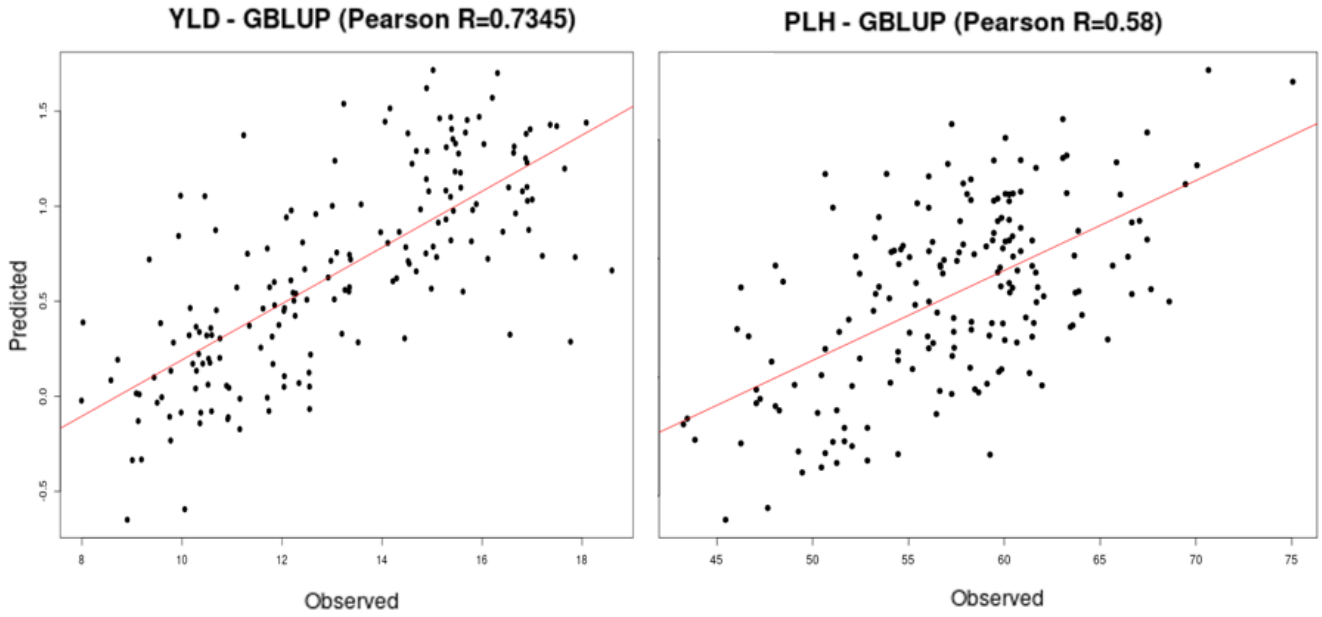

Figure 4. Scatter plot of the observed values versus and GEBVs using GBLUP model for YLD (A) and PLH (B) showing prediction accuracies for across-population prediction.