COMPARING PREDICTION METHODS OF WHEAT GRAIN QUALITY WITH THE AREA UNDER THE RECEIVER OPERATING CHARACTERISTIC CURVES

A Dissertation
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Ying Lin

In Partial Fulfillment of the Requirements
for the Degree of
DOCTOR OF PHILOSOPHY

Major Program:
Statistics

June 2021

Fargo, North Dakota

North Dakota State University
Graduate School

Title

COMPARING PREDICTION METHODS OF WHEAT GRAIN QUALITY WITH THE AREA UNDER THE RECEIVER OPERATING CHARACTERISTIC CURVES

By

Ying Lin

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

DOCTOR OF PHILOSOPHY

SUPERVISORY COMMITTEE:

Dr. Megan Orr
Chair

Dr. Andrew Green

Dr. Rhonda Magel

Dr. Mingao Yuan

Dr. Nonoy Bandillo

Approved:

June 28, 2021
Date

Dr. Rhonda Magel
Department Chair

ABSTRACT

A widely used breeding method is genomic selection, which uses genome-wide marker coverage to predict genotypic values for quantitative traits. Genomic selection combines molecular and phenotypic data in a training population to obtain the genomic estimated breeding values of individuals in a testing population that have been genotyped but not phenotyped. One popular method for this estimation is G-BLUP. To further simplify data collection efforts and costs, we developed models with linear model, Bayesian linear model, K-nearest neighbors, and Random Forest to predict quality traits and compare the predictive ability of this new approach with G-BLUP using Pearson correlation and area under the receiver operating characteristic curve. The goal of this approach is to enable the analysis of large-scale data sets to provide relatively accurate estimates of quality traits without the time and energy consumption of marker analysis. Application of the methods to predict the quality traits for spring wheat breeding data reveals that compared with G-BLUP methods, the proposed methods perform better in loaf volume prediction, perform poorly in flour extraction and bake absorption prediction, and in mixograph prediction, the performance is not bad.

ACKNOWLEDGEMENTS

I am grateful to all of the teachers and students in the Department of Statistics. They have helped me develop solid an academic background.

I would like to express my sincere gratitude and thanks to my advisor Megan Orr who has provided me professional guidance and taught me a great deal about both scientific research and attitude towards work. I would like to express my appreciation to my co-advisor Andrew Green, who has the attitude and the substance of a genius: he continually and convincingly conveyed a spirit of adventure in regard to research and scholarship. Without their guidance and persistent help, this dissertation would not have been possible. I would like to thank Rhonda Magel, Mingao Yuan and Nonoy Bandillo who were willing to participate as my defense committee.

Finally, I would like to thank my parents and friends for encouraging me in all my pursuits and inspiring me to follow my dreams.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GS ..... Genomic selection

RR-BLUP ..... Ridge regression best linear unbiased prediction

GEBVs ..... Genomic-estimated breeding values

TW ..... Test weight

K-NN ..... K-nearest neighbors

ROC ..... Receiver operating characteristic

REML ..... Restricted maximum likelihood

GAUSS ..... Gaussian model

EXP ..... Exponential model

LM ..... Linear model

BAY ..... Bayesian linear model

RF ..... Random forest

SSD ..... Sum of square deviances

SNPs ..... Single nucleotide polymorphisms

SMOTE ..... Synthetic minority oversampling technology

TP ..... True positive

FP ..... False positive

FN ..... False negative

TN ..... True negative

TPR ..... True positive rate

FPR .....False positive rate

FNR .....False negative rate

TNR ..... True negative rate

AUC ..... Area under the ROC curve

CI ..... Confidence interval

PPV ..... Positive predictive value

RMSE ..... Root mean square error

PMT ..... Peak maximum time

BEM ..... Maximum torque

AM ..... Torque 15 seconds before maximum torque

PM ..... Torque 15 seconds after maximum torque

AE..... Aggregation energy

SUE ..... Start-up energy

PE ..... Plateau energy

GPI ..... Gluten performance index

SRC ..... Solvent retention capacity

FABS ..... Farinograph water absorption

PKT ..... Farinograph peak time

STAB..... Farinograph stability

VOL ..... Loaf volume

1. INTRODUCTION

Wheat grain quality is a complex set of traits that play a critical role for wheat producers, end-users and breeders. The functionality and versatility of wheat stem from the elasticity and flexibility of its gluten. Wheat quality can be defined as the ability of wheat grain or flour to meet end-user specific requirements. The wheat seed is a complex structure, and many characteristics can be measured based on the intended application of the wheat kernels. Various phenotypic traits for cereals, flour, dough, and final products must be evaluated to determine the overall quality and best end-use products. Many laboratory tests must be considered to ensure that the candidate wheat varieties meet the quality requirements for a given end-use product. Grain protein, test weight, milling flour extraction, mixograph, loaf volume as well as the bake absorption, are the most popular properties for wheat varieties (Singh and Kent-Jones 2021).

In order to determine the quality characteristics of wheat grains and suitable end-use products, it is necessary to conduct a milling test to grind wheat into flour. Wheat grains are complex and consist of many distinctive parts. The purpose of milling is to separate the flour-like edible endosperm from various bran skins. The chemical composition of wheat is different, so the composition of flour is also different. Since wheat quality testing requires a large amount of flour, it is time-consuming and expensive. Therefore, in many breeding programs, it is usually evaluated as a final performance test (Singh, R. Paul and Kent-Jones 2021). This situation usually leads to promising wheat varieties that cannot be released due to poor quality. Also, the development of any wheat variety with excellent and specialized end-use traits is restricted. Accurate handling and end-use quality prediction models will enable breeding programs to eliminate unacceptable production lines or production lines for specific goals before time and

resources are invested in production lines that fail the final test (Battenfield, Sarah D., et al. 2009).

The goal of this paper is to provide accurate estimates of quality traits without using data that are time consuming or costly to obtain, including marker data and large-scale milling and baking measurements. By using phenotypic data from traits that are easy to measure to predict other traits that are difficult and expensive to obtain, we will identify underperforming lines to discard, and at the same time, advance high-quality lines in the breeding process, thus saving labor, time, and money.

The rest of this paper is organized as follows. In Section 2, we review the literature on genomic selection methods using genotypic data and introduce additional methods with potential for analyzing only phenotypic data to make wheat grain quality predictions. Section 3 presents the methodology of the proposed methods and the nonparametric method used to compare the predictive ability of these methods. Section 4 describes the data sets that were analyzed and used to compare the methods of analysis. Section 5 presents the results of the analysis and comparisons of the different methods. Conclusions and future research are presented in Section 6.

2. LITERATURE REVIEW

A widely used breeding method is genomic selection (GS), which uses genome-wide marker coverage to predict phenotypic values for quantitative traits. Within the scope of quantitative trait breeding, marker-assisted selection with previously identified significant markers has limited ability to predict complex traits (Heffner et al., 2011). However, the genomic selection models use high-density genotype datasets and simultaneously simulate all additive genetic variances. These models use an entry with known phenotypes and genotypes to train the algorithm, cross-validate the predictions and then use only the available genotype information to predict traits in the material. This method which was initially introduced into animal breeding demonstrates that the Ridge Regression and Bayesian methods could be used to simulate total additive variance and predict breeding values (Meuwissen et al., 2001). The claim that getting a genome-wide marker profile will become cheaper than phenotyping each individual is becoming a reality (Poland and Rife, 2012). One study aimed to determine the predictive ability of several GS models for all necessary processing and end-use quality traits, to assess the predictive ability of the forward prediction to the next year, and to introduce end-use quality GS into the CIMMYT bread wheat breeding program (Battenfield et al., 2016).

Emerging research on crop plants suggests that GS may be a handy tool for plant breeding (Heffner et al. 2009). GS combines molecular and phenotypic data in a training population to obtain the genomic estimated breeding values of individuals in a testing population that have been genotyped but not phenotyped. The main advantages of GS over phenotype-based selection in breeding are that it reduces the cost per cycle and the time required for variety development (Lorenzana and Bernardo 2009). Additionally, several authors have used breeding data to study genomic selection extensively and evaluate plant breeding programs by optimizing

training populations (Asoro et al., 2011; Zhao et al., 2013; Liu et al., 2015; Isidro et al., 2015) and genotype * environment interaction (Burgueño et al, 2012; Heslot et al, 2014; Jarquín et al, 2014; Lado et al, 2016).

The ridge regression BLUP (RR-BLUP) method can simultaneously estimate all marker effects for GS (Meuwissen et al., 2001; Whittaker et al., 2000). Rather than categorizing markers as either significant or as having no effect, ridge regression shrinks all marker effects toward zero (Breiman, 1995; Whittaker et al., 2000). The method assumes that markers are random effects with a common variance (Meuwissen et al., 2001). The equal variances assumption does not imply that all markers have the same effect (Bernardo and Yu, 2007) but that marker effects are all equally shrunken toward zero. Nevertheless, the assumption that individual markers have the same variance is unrealistic, and therefore, RR-BLUP incorrectly treats all effects equally (Xu, 2003b). Despite the incorrect assumption of equal marker variance, RR-BLUP is superior to subset selection because it is a very stable procedure in the sense that small changes in the data do not cause large changes in the estimated coefficients, while subset selection is unstable and produces more variable response to selection. (Whittaker et al., 2000).

There is a close connection between marker-based RR-BLUP and G-BLUP, in which the performance of breeding lines is predicted based on their G to other germplasm (Jeffrey B., 2011). The basic G-BLUP model is

$$y = \boldsymbol{W}\boldsymbol{g} + \varepsilon$$

$$\boldsymbol{g} \sim N\left(0, \boldsymbol{K}\sigma_g^2\right),$$

where $\boldsymbol{g}$ is a vector of genotypic values. In pedigree-based prediction of breeding values, K is the additive relationship matrix $\boldsymbol{A}$ derived from the coefficients of coancestry (Bernardo, 2010).

These coefficients reflect the average behavior of alleles undergoing Mendelian segregation, but the actual segregation can be captured with the marker-based relationship matrix

$$\boldsymbol{K}_{RR} = \boldsymbol{G}\boldsymbol{G}'.$$

This equation has the property that, for random populations, its expected value is proportional to $\boldsymbol{A}$ plus a constant (Habier et al., 2007); for this reason, it has been called the realized (additive) relationship matrix. Another key property of $\boldsymbol{K}_{RR}$ is that the genomic-estimated breeding values (GEBVs) it produces are equivalent to those from the marker-based RR-BLUP approach (Hayes et al., 2009).

The above-mentioned genomic selection methods use genome markers to predict wheat quality, but we noticed that some of these qualities are usually correlated with each other. The viscoelasticity of wheat dough is mainly regulated by storage proteins, gluten, and gliadin (Delcour and Hoseney, 2010; Garg et al., 2006; Payne et al., 1987; Zheng et al., 2009). Wheat breeders often use mixograph (National Manufacturing Co., Lincoln, NE) results to screen early generation lines for dough gluten strength. Flour water absorption measured by the mixograph often serves as bake absorption in bread baking tests.

Of particular importance is that different traits need different time and energy to measure. Grain volume weight, or test weight (TW), and grain protein are the first measurements taken if the wheat breeders are conducting a quality test. They are the easiest and cheapest traits to measure compared with other quality traits since they are determined in the wheat and milling test. These parameters are also important because they are measured when producers sell grain, and deficiencies in either can result in cash discounts. To get the phenotypic data of mixograph, loaf volume, water absorption and milling extraction, testing on flour and dough is needed. Flour extraction is the amount of white flour that is extracted from a given weight of clean and

conditioned wheat. The mixograph measures and records resistance of a dough to mixing. (AACC Method 54-40.02).

Thereby, we proposed new methods to predict the quality traits based on these two parameters alone. These methods include traditional and machine learning models with test weight and protein as independent factors, combined with the influence of the year, to predict other quantitative traits and identify underperforming wheat lines. To simplify data collection efforts and costs, we built Bayesian linear model (BAY), k nearest neighbors (K-NN), and Random forest (RF) models without markers to classify some quality traits. The methods we were proposing are more suitable for identifying low quality wheat lines for breeding and are potentially easier because they use only a subset of the phenotypic data. It can reduce the money and energy for the breeders on markers or large-scale milling and baking.

Bayesian linear regression is a linear regression method in which statistical analysis is performed in the context of Bayesian inference (Box, G. E. P.; Tiao, G. C., 1973). When the error of the regression model has a normal distribution, and the specific form of the prior distribution is assumed, the explicit result can be used for the posterior probability distribution of the model parameters. Reference prior distribution on coefficients will provide a connection between the frequentist solutions and Bayesian answers.

K nearest neighbors is often used in search applications where you are looking for "similar" items. For various forest inventory mapping and estimation applications, this technique has become extremely popular. This popularity can be attributed to its non-parametric, multivariate features, intuitiveness, and ease of use (Altman, Naomi S., 1992). This technique can assign weights to the contribution of neighbors. Therefore, whether it is classification or regression, the contribution of the closer neighbors to the mean is greater than that of the distant

neighbors. In this paper, we used K-NN regression, the output of which is the attribute value of the object. This value is the average of the values of the k nearest neighbors. If the features are displayed at very different scales or use different physical units, normalizing the training data can greatly improve its predictive ability because the algorithm relies on distance.

Random forest is one of the newest members of non-parametric statistics and machine learning algorithms. Facts have proved that this technology has excellent performance on many practical problems, but its mechanism has not been fully understood (Ho, Tin Kam, 1995). Random forest is composed of many overall decision trees. Each tree in the random forest will return a category prediction, and the category with the most votes will be the prediction of our model. A large number of relatively unrelated trees operating as a committee will outperform any single component model.

A Cross-validation method using a population with genotypic and phenotypic data is commonly used to assess GS accuracy. This accuracy is theoretically equal to the correlation between predicted phenotypes and observed phenotypes divided by the square root of heritability (Lee et al., 2008). In this article, however, we used another standard as an estimate of the predictive ability, which is based on the ranking of traits values to identify relatively sufficient or underperforming groups by looking at varying selection intensities and determining whether they should be kept based on a simulated breeding situation. This is unique from most of the literature in that it is not a “straight” predictive ability. Since this test is based on an observed variable that lies on a graded scale, we evaluated the overall value of the test by using receiver operating characteristic (ROC) curves (Hanley and Mcneil, 1982; Metz, 1978). The curve will then pass through the point (0,1) on the unit gird. The closer the ROC curve comes to the ideal point, the

better its discriminating ability. In this approach, the population is divided into training set and test set.

The area under the ROC curve represents a recommended index of predictive ability. It is a useful indicator of both the magnitude or importance of a difference between two populations and the predictive ability of discrimination performance. The usual estimator for this area is closely related to the Mann-Whitney U statistic. We used some properties of this nonparametric statistic to compare areas under ROC curves arising from proposed methods and G-BLUP applied to the same individuals. This approach calculates the correlation between the original measurements. The average of the two correlations is used along with the average of the areas under the two curves to derive an estimated correlation between the two areas (Hanley and Mcneil, 1983).

3. METHODOLOGY

Genomic selection, or the prediction of GEBVs using dense molecular markers, is rapidly emerging as a key component of efficient breeding programs. The basic linear regression model used to predict GEBVs with regularization models is:

$$\boldsymbol{y} = \mu\mathbf{1}_{\boldsymbol{n}} + \boldsymbol{X\beta} + \boldsymbol{e},$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^T$ is the vector of observed phenotypes, $\mathbf{1}_{\boldsymbol{n}}$ is a column vector of n ones, μ is a common intercept, $\boldsymbol{X}$ is a $n \times p$ matrix of markers; $\boldsymbol{\beta}$ is the vector of the regression coefficients of the markers and $\boldsymbol{e}$ is the vector of the residual errors with $var(\boldsymbol{e}) = \boldsymbol{I}\sigma_e^2$. In what follows, we assume that the observed phenotypes have been mean-centered.

3.1. Data Set Descriptions

3.1.1. Original data set

The original dataset, which was the motivation for this research, contains both phenotypic and genotypic data from 466 wheat lines from the North Dakota State University spring wheat breeding program. This advanced yield trials are grown in 4 - 7 diverse locations across North Dakota. The trials were replicated but quality samples were taken from single representative plots. The trials were managed by Hard Red Spring Wheat breeding project and Research Extension Center Agronomists. Management was in accordance with local practices.

The genotypic data from 16383 single nucleotide polymorphisms (SNPs) were obtained using genotyping-by-sequencing (GBS). Briefly, DNA was isolated with the Wizard Genomic DNA Purification Kit (A1125; Promega) per the manufacturer's instructions and quantified with a Quant-iT PicoGreen dsDNA assay kit (P7589; Thermo Fisher Scientific). GBS libraries were constructed based on the protocol of Poland et al. (2012) with minor modifications.

The phenotypic data is comprised of six quality trait variables, including two predictors: TW and protein, and four response variables: loaf volume, bake absorption, flour extraction, and mixograph classification. By considering genotype as a random effect, best linear unbiased predictors (BLUPs) across all locations were used for prediction analysis. Single values from each line at each environment (location by year combination) were used to get BLUP value for each line.

Of the 466 wheat lines, the bottom 15% of the phenotypic data for each year were considered wheat lines which should be removed from the breeding program based on their quality phenotype; the remaining are considered successes. N is the number of total wheat lines from 2011 to 2018 year; M is the number of lines falling below the 15% cutoff. The number of wheat lines varies because some lines were tested in multiple years and only unique lines are shown.

Table 1. The number of unique spring wheat genotypes for each year considered

YEAR	**2011**	**2012**	**2013**	**2014**	**2015**	**2016**	**2018**
N	119	80	34	76	49	61	47
M	30	20	9	19	13	16	12

3.1.2. New data set

We also applied a subset of the proposed methods to a new data set containing more phenotypic predictor variables but no genotypic data. This dataset utilized 48 hard red spring wheat genotypes grown in 2018 and 2019, and these genotypes were sent to the North Dakota State University bread wheat quality laboratory (PI=Dr. Senay Simsek) for flour, dough, and baking quality tests. It includes peak maximum time (PMT), maximum torque (BEM), torque 15 seconds before and after maximum torque (abbreviated as AM and PM, respectively), aggregation energy (AE), start-up energy (SUE), plateau energy (PE), gluten performance index

(GPI), solvent retention capacity (SRC), farinograph water absorption (FABS), farinograph peak time (PKT), farinograph stability (STAB) and loaf volume (VOL).

FABS is an important factor in determining the quality of final products, and flour samples with high water absorption are more suitable for making bread (Ma et al., 2007). STAB measured the length of time the dough maintains maximum consistency. PKT was recorded as the time from the addition of water to the flour until the dough reaches its maximum consistency.

The SRC was measured as the weight of the solvent remaining in the flour after centrifugation. It was expressed as a percentage of the moisture content of the flour sample at 14%. GPI is the ratio of lactic acid SRC to the sum of SRC values. Gluten aggregation properties of flour samples were also measured to get some quality trait variables. PMT was measured as the time in seconds to reach maximum torque. BEM measured the maximum resistance of gluten to mixing. AE was measured as the area under the curve between 15 seconds before and after the maximum torque (AACCI Method 2010).

3.2. Traditional Method: G-BLUP

Ridge regression (RR) is ideal if there are many predictors, all with non-zero coefficients and drawn from a normal distribution. It performs well with many predictors, each having a small effect, and prevents coefficients of linear regression models with many correlated variables from being poorly estimated and exhibiting high variance. RR shrinks the coefficients of correlated predictors equally towards zero. For example, given k identical predictor variables, each variable will get the same coefficient, which is equal to 1/k of the size that a single predictor variable will get when it is fitted separately (Friedman J, Hastie T, Tibshirani R 2010). Therefore, RR does not force the coefficients to zero, so it is impossible to choose a model with only the most relevant and predictive subset of predictions.

The ridge regression estimator solves the regression problem using ℓ2 penalized least squares:

$$\widehat{\boldsymbol{\beta}}(ridge) = \underset{\boldsymbol{\beta}}{argmin} \sum_{i=1}^{n}(y_i - x_i\beta)^2 + \lambda\|\boldsymbol{\beta}\|_2^2,$$

where x_i^T is the ith row of $\boldsymbol{X}$, $\|\boldsymbol{\beta}\|_2^2 = \sum_{j=1}^{p}\beta_j^2$ is the ℓ2 -norm penalty on $\boldsymbol{\beta}$, and $\lambda \geq 0$ is the tuning parameter which regulates the strength of the linear shrinkage by determining the relative importance of the data-dependent empirical error and the penalty term. The intercept is assumed to be zero in the $\widehat{\boldsymbol{\beta}}$ estimator due to mean-centering of the phenotypes. RR-BLUP uses the same estimator as ridge regression but estimates the penalty parameter by the restricted maximum likelihood (REML) as $\lambda = \sigma_e^2/\sigma_\beta^2$, where σ_e^2 is the residual variance, and $var(\widehat{\boldsymbol{\beta}}) = \boldsymbol{I}\sigma_\beta^2$ is the variance matrix of the regression coefficients (Piepho H.P. 2009).

The marker-based RR-BLUP and G-BLUP have a close connection, in which the performance of breeding lines is predicted based on their G to other germplasm (Jeffrey B. Endleman, 2011). The basic G-BLUP model is

$$\boldsymbol{y} = \boldsymbol{Z\mu} + \boldsymbol{X\beta} + \boldsymbol{\varepsilon}, \boldsymbol{\mu} \sim N(0, \boldsymbol{K}\sigma_u^2),$$

where $\boldsymbol{X}$ is a full-rank design $n \times p$ matrix of markers for the fixed effects $\boldsymbol{\beta}$, $\boldsymbol{Z}$ is the design matrix for the random effects $\boldsymbol{\mu}$, $\boldsymbol{K}$ is a positive semidefinite matrix, and the residuals are normally distributed with constant variance. Variance components are estimated by REML using the spectral decomposition algorithm of Kang et al. (2008).

When the realized relationship matrix

$$\boldsymbol{K} = \boldsymbol{GG}'$$

is used, the RR-BLUP and G-BLUP of the prediction problem give equivalent GEBVs. G is the genotype matrix (e.g., {0,1} for biallelic single nucleotide polymorphisms (SNPs) under an

additive model). The realized relationship model is in fact a kernel in genotype space and can be written as

$$K_{ij} = \langle G_{i.}, G_{j.} \rangle,$$

where the angle brackets denote the inner product between genotypes i and j. In the additive relationship model, the genetic covariance between lines is proportional to their similarity in the genotype space, because the inner product measures the geometric similarity of two vectors (Jeffrey B. Endelman 2011).

There are two kernels other than RR. One is the Gaussian model (GAUSS):

$$K_{ij} = \exp\left[-\left(\frac{D_{ij}}{\theta}\right)^2\right],$$

where

$$D_{ij} = \left[\frac{1}{4M}\sum_{k=1}^{M}\left(G_{ik} - G_{jk}\right)^2\right]^{\frac{1}{2}}$$

is the Euclidean distance between genotypes i and j, normalized to the interval [0,1]. Endelman defined the parameter θ as a proportional parameter that affects the speed at which the genetic covariance decays with the distance (2011). The other kernel is the exponential model (EXP):

$$K_{ij} = \exp\left[-\frac{D_{ij}}{\theta}\right].$$

These kernels are available through the rrBLUP function G.BLUP in the R software, which aim to predict the genotype value of one population based on the genotype and phenotype of the second trained population.

3.3. Linear Regression

What follows is a simple but essential model that will be the basis for a later study of predicting quality traits. First, a random variable Z has a standard normal distribution if its

density function $f_Z(z)$ is given by the standard normal density function $\emptyset(\mathrm{z}) = \frac{e^{-z^2/2}}{\sqrt{2\pi}}$. The function $\Phi(\mathrm{z}) = \int_{-\infty}^{z} \emptyset(u)du$ denotes the distribution function of a standard normal variable, so an equivalent condition is that the distribution function of Z satisfies $F_Z(z) = P(Z \leq z) = \Phi(\mathrm{z})$.

A random variable Y is normal (u, σ^2). Note that the parameters u and σ are the mean and standard deviation of Y, respectively. The parameter μ affects the location, and the parameter σ effects the spread of a normal distribution.

We used phenotypic data in a training population to obtain estimated breeding values of individuals in a test population. Let independent variables x_1 and x_2 denote the genotypes of TW and protein to predict other traits in the same year. The regression linear models (LM) are created as following:

- Loaf Volume Prediction: $y_{1i} = \beta_{10} + \beta_{11}x_1 + \beta_{12}x_2 + \beta_{1i} + \beta_{13}x_1x_2 + \varepsilon_1$, where y_{1i} is the value of loaf volume, β_{11} is the parameter of TW on loaf volume, β_{12} is the parameter of protein on loaf volume, β_{13} is the interaction coefficient of two predictors.
- Mixograph Prediction: $y_{2i} = \beta_{20} + \beta_{21}x_1 + \beta_{22}x_2 + \beta_{2i} + \beta_{23}x_1x_2 + \varepsilon_2$, where y_2 is the value of mixograph, β_{21} is the parameter of TW on mixograph, β_{22} is the parameter of protein on mixograph, β_{23} is the interaction of two predictors.
- Flour Extraction Prediction: $y_{3i} = \beta_{30} + \beta_{31}x_1 + \beta_{32}x_2 + \beta_{3i} + \beta_{33}x_1x_2 + \varepsilon_3$, where y_3 is the value of flour extraction, β_{31} is the parameter of TW on flour extraction, β_{32} is the parameter of protein on flour extraction, β_{33} is the interaction of two predictors.
- Bake Absorption Prediction: $y_{4i} = \beta_{40} + \beta_{41}x_1 + \beta_{42}x_2 + \beta_{4i} + \beta_{43}x_1x_2 + \varepsilon_4$, where y_3 is the value of absorption, β_{41} is the parameter of TW on bake absorption, β_{42} is the parameter of protein on bake absorption, β_{43} is the interaction of two predictors.

3.4. Bayesian Linear Regression

Since the interaction of TW and protein is not significant for all response variables, we only considered the two predictors of TW and protein when developing the Bayesian linear regression model in order to simplify the formula. Under the assumption that the errors ε are normally distributed with constant variance σ^2, we have for the random variable of each response y_i (loaf volume, flour extraction, mixograph or bake absorption),

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon = \boldsymbol{\beta}^T \boldsymbol{X} + \varepsilon$$

conditioned on the observed data $\boldsymbol{X}$ and the parameters $\boldsymbol{\beta}, \sigma^2$, is normally distributed as

$$y_i|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2 \sim N(\boldsymbol{\beta}^T \boldsymbol{X}, \sigma^2).$$

Let $D = (y_1, \ldots, y_n)$ be the data. The likelihood function is

$$f(D|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = \prod_{i=1}^{n} f(y_i|\boldsymbol{X}, \boldsymbol{\beta}, \sigma^2) = (2\pi\sigma^2)^{-n/2} \exp\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^T \boldsymbol{X})^2\}.$$

We first considered the case under the reference prior, which is a standard noninformative prior. Using this reference prior, we obtained distributions as the posterior distributions of $\boldsymbol{\beta}, \sigma^2$. The non-informative prior distribution was set $\pi(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{\sigma^2}$. Using the hierarchical model framework, this is equivalent to assuming that the joint prior distribution of $\boldsymbol{\beta}$ under σ^2 is a uniform prior, while the prior distribution of σ^2 is proportional to $\frac{1}{\sigma^2}$. That is, $\pi(\boldsymbol{\beta}|\sigma^2) \propto 1$ and $\pi(\sigma^2) \propto \frac{1}{\sigma^2}$.

Then we applied Bayes' rule to derive the joint posterior distribution of $\boldsymbol{\beta}, \sigma^2$, which is proportional to the product of the likelihood and the joint prior distribution:

$$f(\boldsymbol{\beta}, \sigma^2|D) \propto (\sigma^2)^{-\frac{n}{2}-1} \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \boldsymbol{\beta}^T \boldsymbol{X})^2\right\}.$$

So the posterior distribution of $\boldsymbol{\beta}$ conditioning on σ^2 is $\boldsymbol{\beta}|\sigma^2, D \sim t\left(n-3, \widehat{\boldsymbol{\beta}}, \left(se_{\boldsymbol{\beta}}\right)^2\right)$, and the posterior distribution of σ^2 is $\frac{1}{\sigma^2} \sim Gamma\left(\frac{n-3}{2}, \frac{SSE}{2}\right)$, where $SSE = \sum_i^n (y_i - \hat{y}_i)^2$. The posterior mean, $\widehat{\boldsymbol{\beta}}$, is the center of the t-distribution of $\boldsymbol{\beta}$, which is the same as the frequentist ordinary least square estimates of $\boldsymbol{\beta}$. The standard errors $se_{\boldsymbol{\beta}}$ are same as the ordinal least squares' estimates, given as

$$se_{\beta_0} = \hat{\sigma}\sqrt{\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}}}$$

$$se_{\beta_1} = se_{\beta_2} = \sqrt{\frac{\hat{\sigma}^2}{S_{xx}}}.$$

3.5. K Nearest Neighbors

The k nearest neighbors method (K-NN) is a non-parametric algorithm that stores all available cases and predicts the numerical target based on a similarity measure (e.g., distance functions). The method takes many labelled points and uses them to learn how to label other points. When an unknown vector is to be classified, its *k* closest neighbors are found from among all the prototype vectors, and the class label is decided based on a majority rule. To avoid ties on overlap regions, the value of *k* should be odd.

A simple implementation of K-NN regression is to calculate the average of the numerical target of the K nearest neighbors. A common selection for the distance metric is Euclidean distance, $d_{ij} = \sqrt{\left(x_{1i} - x_{1j}\right)^2 + \left(x_{2i} - x_{2j}\right)^2}$ where *i* and *j* denote individuals, x_{1i} and x_{1j} are the values of TW for the ith and jth individual, x_{2i} and x_{2j} are the values of protein for the ith and jth individual.

We choose *k* through cross-validation by partitioning the dataset into training data, cross-validation data, and test data. We then used the training data for finding the nearest neighbors, the cross-validation data to find the best value of *k*, and finally the test data to test our model. In this way, we used different values of *k* to predict the label of each individual in the validation set, and then used that value in the final setup of the algorithm.

The K-NN prediction for the ith individual is $\hat{y}_i = \frac{\sum_{j=1}^{k} y_{ij}}{k}$, where i denote individuals and j denote neighbors of individual i, y_{ij} is the response value (loaf volume, mixograph, flour extraction or bake absorption) for the ith individual.

3.6. Random Forest

A set of T unpruned ensemble of regression trees constitutes the random forest. These trees are generated based on bootstrap sampling in the original training data. The bootstrap resampling of the data used to train each tree can increase the diversity between trees. Each tree is composed of root nodes, branch nodes, and leaf nodes. For each node of the tree, the optimal node is selected to separate features from the set of features, which are randomly selected from the feature space (Rahman, R., Dhruba, S.R., Ghosh, S. et al. 2019).

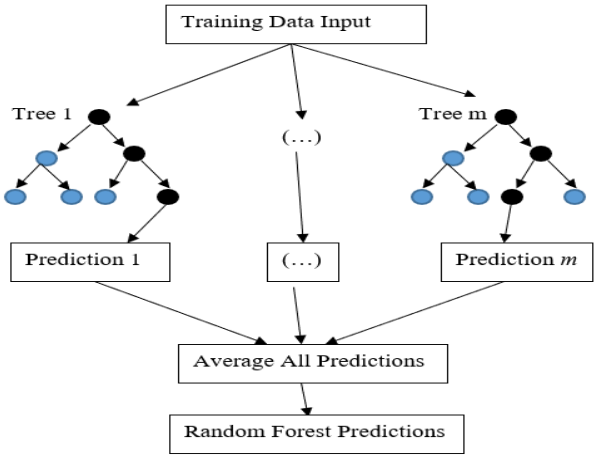

Figure 1. Random Forest algorithm structure

The figure above shows the structure of a random forest. These trees run in parallel and there is no interaction between them. The operation of random forest is to construct several decision trees during training and then output the average value of the class as the prediction of all trees. Using the randomized feature selection process, we fit the tree based on bootstrap samples $\{(X_1, Y_1), (X_2, Y_2), \dots, (X_m, Y_m)\}$ from training data. We performed the following three steps:

1) Randomly selected data points from the training set.
2) Built a decision tree related to these data points.
3) Chose the number *m* of trees, and repeated steps 1 and 2.
4) For a new data point from the test data, made each tree predict the response value of that data point, and assigned the new data point to the average of all predicted response values.

The averaging makes the random forest better than a single decision tree, thus improving its predictive ability and reducing overfitting.

3.7. Data Treatment

3.7.1. Best linear unbiased prediction

By treating the genotype as a random effect, the best linear unbiased predictor (BLUP) was used in this study for predictive analysis. BLUP was derived from Charles Roy Henderson in 1950. BLUP is a technique for estimating genetic value. Generally speaking, it is a method of estimating random effects. It can be used to remove noise in the image and estimate small areas.

The model for quality traits observations $\{Y_j, j = 1, \dots, n\}$ is written as

$$Y_j = u + x_j^T \beta + \xi_j + \varepsilon_j,$$

where ξ_j and ε_j represent the random effect and observation error for observation j. Suppose ξ_j and ε_j are uncorrelated and have known variances σ_ξ^2 and σ_ε^2, respectively. Further x_j is a vector

of independent variables for the jth observation and β is a vector of regression parameters. The BLUP of kth observation can be defined as $\hat{Y}_k = \sum_{j=1}^{n} c_{j,k} Y_j$, where $c_{j,k}$ is the linear coefficient of the predictor $\hat{Y}_k$. This BLUP was chosen to minimize the variance of the prediction error

$$Var(u + x_k^T \beta + \xi_k - \hat{Y}_k)$$

and was subject to the condition of unbiased predictor variables

$$E\left(u + x_k^T \beta + \xi_k - \hat{Y}_k\right) = 0.$$

3.7.2. Steps for group definition

To clarify the definition of the groups, we performed the following three steps:

1) Obtain predicted values by the G-BLUP models and proposed method models.
2) Rank predicted values and assign those in the bottom (least desirable) b% as group 1 and others as group 2. We used b=15.
3) Rank observed values and assign those in the bottom b% as group 1 and others as group 2.

3.7.3. Cross-validation for time series predictor evaluation

We used block cross-validation and sufficient control over stationarity because it makes full use of all available information for training and testing, resulting in reliable error estimates (Christoph and José 2012).

Our dataset is a collection of observations obtained through same wheat lines over time, so it is time series ordered data. Time-series ordered data can be problematic for cross-validation, so we used a forward-chaining approach that is sometimes more appropriate for time series, where the procedure is as follows (Ma, Z., Dai, Q. 2016):

fold 1: training [2011], test [2012]

fold 2: training [20111, 2012], test [2013]

fold 3: training [2011, 2012, 2013], test [2014]

fold 4: training [2011, 2012, 2013, 2014], test [2015]

fold 5: training [2011, 2012, 2013, 2014, 2015], test [2016]

fold 6: training [2011, 2012, 2013, 2014, 2015, 2016], test [2018]

This approach more accurately models the situation we see at prediction time, where we model on past data and predict on forward-looking data.

3.7.4. Synthetic data generation for imbalanced data

When we divided wheat lines into sufficient and underperforming groups, the number of lines in the underperforming group (group 1) was much less than the number of lines in the sufficient group (group 2). Therefore, this imbalance in the data must be considered prior to building predictive models. Regarding the generation of synthetic data, synthetic minority oversampling technology (SMOTE) is a powerful and widely used method. The SMOTE algorithm creates artificial data based on the similarity of the TW and protein of a few samples. It generates a set of random observations in the underperforming group to bias the learning of the classifier to the underperforming group. This technique was proposed by Chawla, Bowyer, Hall, and Kegelmeyer in 2002, and has become an established method, extending to more than 85 kinds of basic methods. One way to visualize how basic concepts work is to imagine drawing a line between two existing minority data points. Then, SMOTE creates a new synthetic instance somewhere between these rows.

To generate artificial data, bootstrapping and k-nearest neighbors are implemented. Precisely, it works this way:

1) Take the difference between the feature vector under consideration and its nearest neighbor.
2) Multiply this difference by a random number between 0 and 1.
3) Add it to the feature vector under consideration.

4) This causes the selection of a random point along the line segment between two specific features.

3.8. Traditional Predictive Ability

Model evaluation is the task of evaluating a model, which is essential for selecting the best model among several possible choices. In quantitative genetics, it is common to use the Pearson correlation coefficient between predicted values and BLUPs of phenotypic values as a model selection criterion (González-Recio et al., 2014). It theoretically represented by

$$r_{y\hat{y}} = \frac{COV[y, \hat{y}]}{\sqrt{VAR[y]VAR[\hat{y}]}},$$

where $COV[y, \hat{y}]$ is covariance of observed and predicted values, $VAR[y]$ is the variance of observed values, and $VAR[\hat{y}]$ is the variance of predicted values. The model is better when $r_{y\hat{y}}$ is higher and perfect correlation is achieved when $r_{y\hat{y}} = 1$.

3.9. Mann-Whitney Two-sample Test to Compare the AUC

After getting the predicted value and prediction of group label of loaf volume, mixograph, and flour extraction based on the effect of TW, protein, and their interaction, we checked the predictive ability of these models. If the observed value of a wheat line predicted to be in group one falls in the lowest b% of observed values (we used b=15), then this wheat line was correctly classified, and is considered a true positive (TP). Simply stated, the variable that is in the bottom b% group and is predicted to be in the bottom b% is TP; the variable that is not in the bottom b% group and is predicted to be in the $1 - $ b% group is true negative (TN). The following table illustrate true positive rate (TPR), false positive rate (FPR), false negative rate (FNR), and true negative rate (TNR)

Table 2. True positive rate and False positive rate

		TRUE CONDITION	
		Bottom b% (P)	$1-$ b% group (N)
PREDICTED CONDITION	Predicted bottom b%	True positive (TP)	False positive (FP)
	Predicted $1-$ b% group	False negative (FN)	True negative (TN)
		TPR=TP/P FNR=FN/P	FPR=FP/N TNR=TN/N

Receiver operating characteristic curves (ROC curves) are created by plotting the TPR against the FPR at various threshold settings. ROC curves are an excellent way to see how any predictive model can distinguish between true positives and negatives. The best predictive model should have a high TPR and a low FPR.

Suppose a sample of N individuals undergo a test for predicting phenotypic value by two methods. The observed values of the m of these individuals are in the bottom b% group. Let this group be denoted by C_1 and let the group of $\mathrm{n} = \mathrm{N} - \mathrm{m}$ individuals who are within the $1 - \mathrm{b}\%$ group be denoted by C_2. Let $X_i, i = 1,2, \dots, m$ and $Y_j, j = 1,2, \dots, n$ be the indicator values of the variable on which the diagnostic test is based for members of C_1 and C_2, to construct respectively.

$$X_i = \begin{cases} 1, i^{th}\ individual\ of\ observed\ values\ is\ in\ the\ C_1 \\ 0, i^{th}\ individual\ \ of\ observed\ values\ is\ in\ the\ C_2 \end{cases}$$

$$Y_j = \begin{cases} 1, j^{th}\ individual\ of\ predicted\ values\ is\ in\ the\ C_1 \\ 0, j^{th}\ individual\ of\ predicted\ values\ is\ in\ the\ C_2 \end{cases}$$

These results can be used to construct an empirical ROC curve for assessing the diagnostic performance of the test. Let $\mathrm{TPR} = \frac{1}{m}\sum_{i=1}^{m} I(X_i = Y_j = 1)\ (j = 1,2, \dots, n)$ where $I(A) = 1$ if A is true and 0 otherwise. Also let $\mathrm{FPR} = \frac{1}{n}\sum_{j=1}^{n} I(X_i = Y_j = 0)\ (i = 1,2, \dots, m)$.

Then, TPR is the empirical sensitivity of the test, which is derived by dichotomizing the variable into positive or negative results, and FPR is the corresponding empirical specificity. It has been shown that the area under an empirical ROC curve (AUC) is equal to the Mann-Whitney two-sample statistic applied to the two samples C_1 and C_2 when calculated by the trapezoidal rule. Since the Mann-Whitney statistic is a generalized U-statistic, statistical analysis of diagnostic tests performance can be performed by using the general theory of U-statistics.

The Mann-Whitney statistic estimates the probability, θ, that a predicted class label will be same as the observed group label. It can be computed as the average over a kernel φ, as $\hat{\theta} = \frac{1}{mn}\sum_{j=1}^{n}\sum_{i=1}^{m}\varphi(X_i, Y_j)$, where $\varphi(X_i, Y_j) = \begin{cases} 1, Y_j = X_i \\ 0, Y_j \neq X_i \end{cases}$.

Sen (1960) has provided a method of structural components to provide consistent estimates of the elements of the variance-covariance matrix of a vector of U-statistics. This method is equivalent to jackknifing but is conceptually more straightforward when dealing with U-statistics. We exploit this methodology to compare the areas under two ROC curves. For the proposed methods, $\hat{\theta}^s$, the X-components and Y-components are defined, respectively, as

$$V_{10}^{s}(X_i) = \frac{1}{n}\sum_{j=1}^{n}\varphi\left(X_i^s, Y_j^s\right)\ (i = 1,2,\dots,m)$$

and

$$V_{01}^{s}(X_i) = \frac{1}{m}\sum_{i=1}^{m}\varphi\left(X_i^s, Y_j^s\right)\ (j = 1,2,\dots,n).$$

For the G-BLUP approach, set

$$V_{10}^{r}(X_i) = \frac{1}{n}\sum_{j=1}^{n}\varphi\left(X_i^r, Y_j^r\right)\ (i = 1,2,\dots,m)$$

and

$$V_{01}^{r}(X_i) = \frac{1}{m}\sum_{i=1}^{m} \varphi\left(X_i^r, Y_j^r\right)\ (j = 1,2, \dots, n).$$

Also define the 2×2 matrix S_{10} such that the elements are:

$$s_{10}^{s,r} = s_{10}^{r,s} = \frac{1}{m-1}\sum_{i=1}^{m}\left[V_{10}^{s}(X_i) - \hat{\theta}^{s}\right]\left[V_{10}^{r}(X_i) - \hat{\theta}^{r}\right],$$

$$s_{10}^{s,s} = \frac{1}{m-1}\sum_{i=1}^{m}\left[V_{10}^{s}(X_i) - \hat{\theta}^{s}\right]^2,$$

$$s_{10}^{r,r} = \frac{1}{m-1}\sum_{i=1}^{m}\left[V_{10}^{r}(X_i) - \hat{\theta}^{r}\right]^2.$$

Similarly define S_{01}, which has elements.

$$s_{01}^{s,r} = s_{01}^{r,s} = \frac{1}{n-1}\sum_{i=1}^{n}\left[V_{01}^{s}(X_i) - \hat{\theta}^{s}\right]\left[V_{01}^{r}(X_i) - \hat{\theta}^{r}\right],$$

$$s_{01}^{s,s} = \frac{1}{n-1}\sum_{i=1}^{n}\left[V_{01}^{s}(X_i) - \hat{\theta}^{s}\right]^2,$$

$$s_{01}^{r,r} = \frac{1}{n-1}\sum_{i=1}^{n}\left[V_{01}^{r}(X_i) - \hat{\theta}^{r}\right]^2.$$

The estimated covariance matrix for the vector of parameter estimates, $\widehat{\boldsymbol{\theta}} = (\hat{\theta}^{s}, \hat{\theta}^{r})$**,** is thus

$$\mathrm{S} = \frac{1}{m}S_{10} + \frac{1}{n}S_{01}.$$

For contrast $\mathrm{L}\boldsymbol{\theta}'$**,** where $\mathrm{L} = (1, -1)$,

$$\frac{\mathrm{L}\widehat{\boldsymbol{\theta}}' - \mathrm{L}\boldsymbol{\theta}'}{[LSL']^{1/2}}$$

has a standard normal distribution. A two-sided 95% confidence interval for $\mathrm{L}\boldsymbol{\theta}'$ is thus

$\left[\mathrm{L}\widehat{\boldsymbol{\theta}}' \pm 1.96[LSL']^{1/2}\right]$.

4. RESULTS

The first analysis used the original data set and compares the linear model, Bayesian linear model, K-nearest neighbors, and random forest methods with G-BLUP methods (ridge regression, exponential, gaussian) on wheat lines from 2011-2018 spring wheat to the best linear unbiased prediction (BLUP) data.

Predictive ability is expressed as the Pearson correlation between predicted values and the true observed values (Table 3). Compared with G-BLUP methods, we can see that linear model, Bayesian linear model, K-nearest neighbors, and random forest perform better in loaf volume prediction, perform poorly in flour extraction and bake absorption prediction, and in mixograph prediction, the performance is not bad. To obtain further conclusions about the accuracy of these predictive models, we proposed a hypothesis to test whether their areas under the ROC curve are equal.

Table 3. Predictive ability of seven models for 4 response variables

YEAR	METHOD	VOLUME	EXTRACTION	MIXOGRAPH	ABSORPTION
2012	BAY	0.47	0.05	0.26	0.13
	LM	0.47	0.05	0.26	0.13
	K-NN	0.47	0.02	0.18	0.09
	RF	0.44	0.14	0.22	0.06
	RR	0.49	0.29	0.32	0.49
	EXP	0.47	0.34	0.26	0.48
	G	0.47	0.33	0.24	0.48
2013	BAY	0.66	0.31	0.24	0.06
	LM	0.67	0.16	0.20	0.08
	K-NN	0.69	0.11	0.27	0.08
	RF	0.66	0.18	0.18	0.11
	RR	0.52	0.55	0.33	0.62
	EXP	0.56	0.50	0.35	0.58
	G	0.53	0.51	0.35	0.59
2014	BAY	0.62	0.26	0.50	0.26
	LM	0.62	0.20	0.52	0.24
	K-NN	0.61	0.20	0.49	0.25
	RF	0.59	0.07	0.48	0.27
	RR	0.66	0.56	0.45	0.37
	EXP	0.64	0.57	0.46	0.36
	G	0.65	0.57	0.48	0.36
2015	BAY	0.71	0.21	0.30	0.40
	LM	0.71	0.21	0.32	0.40
	K-NN	0.68	0.20	0.26	0.49
	RF	0.67	0.28	0.15	0.46
	RR	0.49	0.23	0.41	0.28
	EXP	0.52	0.24	0.39	0.28
	G	0.50	0.25	0.38	0.29
2016	BAY	0.20	0.39	0.40	0.12
	LM	0.20	0.38	0.42	0.12
	K-NN	0.22	0.27	0.26	0.26
	RF	0.17	0.18	0.28	0.12
	RR	0.21	0.17	0.26	0.42
	EXP	0.22	0.23	0.29	0.45
	G	0.21	0.22	0.27	0.42
2018	BAY	0.28	0.04	0.14	
	LM	0.29	0.23	0.28	
	K-NN	0.31	0.18	0.06	
	RF	0.27	0.05	0.11	
	RR	0.04	0.17	0.11	
	EXP	0.08	0.16	0.16	
	G	0.06	0.17	0.13	

Note: BAY = Bayesian linear model; LM = linear model; K-NN = K nearest neighbor; RF = random forest; RR = ridge regression; EXP = exponential; G = Gaussian.

4.1. Linear Regression

We estimated the areas under the step-up ROC curves to check the predictive ability intuitively for each year. We prefer to interpret the AUC as an average TPR across FPR, because $\text{AUC} = \int_0^1 ROC(t)dt$. A perfect classifier has $\text{AUC} = 1$, whereas one that performs no better than chance has an AUC of 0.5.

Next, we used the nonparametric approach described in section 3.8 to compare AUCs for the loaf volume prediction results with LM and G-BLUP. The resulting P-values and two-sided 95% confidence intervals (CI) are presented in Table 4. There is no significant difference between LM over G-BLUP in the year 2012, 2013, 2014 and 2016. The confidence interval is the difference between LM and G-BLUP. Therefore, when CI is positive, it indicates that LM is better; otherwise, G-BLUP is better. Additionally, LM obtained more accurate predictions in 2015 and 2018. Figure 2 shows the ROC curves corresponding to two tests with significant differences. The blue line is the ROC curve of LM, and other lines are the ROC curves of G-BLUP. The area under the blue line is significantly larger than the area under the other lines. These results indicate that LM performed better than G-BLUP in 2015 and 2018.

Considering these data, LM performs similarly or better than G-BLUP when predicting loaf volume. This is because the predictors (TW and protein) are all significant for the linear model method in predicting response variable (loaf volume).

Even if LM is better than G-BLUP, in terms of prediction, it may still be too low to be of any practical value. The first ROC curve in 2015 corresponds to a better prediction than the second in 2018. Therefore, in order to prove whether LM is a good prediction method, we also considered the traditional predictive ability (correlation) and TPR/PPV.

Table 4. LM VS G-BLUP: P-values and 95% CIs for loaf volume prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.22	0.02	0.09	Accept	
	EXP	-0.21	0.03	0.13	Accept	
	G	-0.21	0.03	0.13	Accept	
2013	RR	-0.37	0.03	0.10	Accept	
	EXP	-0.37	0.02	0.08	Accept	
	G	-0.37	0.03	0.10	Accept	
2014	RR	-0.12	0.09	0.77	Accept	
	EXP	-0.13	0.09	0.72	Accept	
	G	-0.13	0.08	0.58	Accept	
2015	RR	0.07	0.28	0.00	Reject	LM
	EXP	0.05	0.27	0.01	Reject	LM
	G	0.06	0.27	0.00	Reject	LM
2016	RR	-0.20	0.04	0.18	Accept	
	EXP	-0.23	0.00	0.06	Accept	
	G	-0.22	0.02	0.09	Accept	
2018	RR	0.17	0.55	0.00	Reject	LM
	EXP	0.15	0.53	0.00	Reject	LM
	G	0.16	0.54	0.00	Reject	LM

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

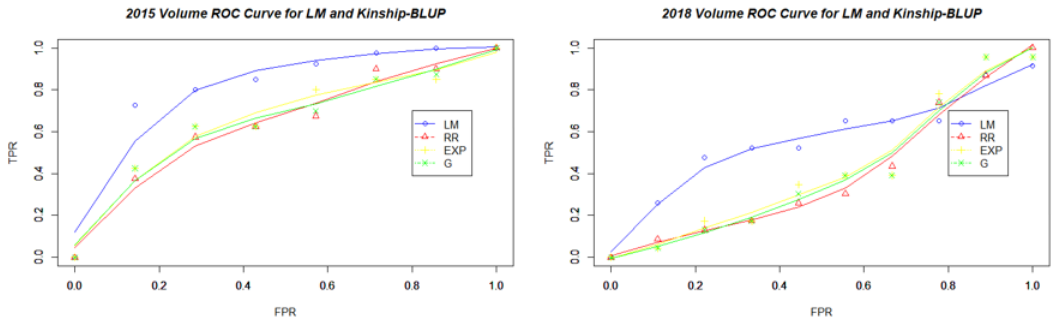

Figure 2. ROC curves of LM and G-BLUP for loaf volume in 2015 and 2018

In Table 5, the AUC results for predicting flour extraction are presented. There is no significant difference between LM and G-BLUP in 2012 and 2018. G-BLUP provides better results based on its AUC in 2013, 2014 and 2016, and LM performs better in 2015(Figure 3). The predictive ability of the flour extraction LM is lower than that of the loaf volume LM, because the predictor TW is not highly related with the response flour extraction in the linear model.

Table 5. LM VS G-BLUP: P-values and 95% CIs for flour extraction prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.23	0.08	0.35	Accept	
	EXP	-0.24	0.06	0.24	Accept	
	G	-0.24	0.07	0.28	Accept	
2013	RR	-0.64	-0.23	0.00	Reject	RR
	EXP	-0.64	-0.19	0.00	Reject	EXP
	G	-0.62	-0.16	0.00	Reject	G
2014	RR	-0.70	-0.37	0.00	Reject	RR
	EXP	-0.69	-0.35	0.00	Reject	EXP
	G	-0.69	-0.37	0.00	Reject	G
2015	RR	0.05	0.42	0.01	Reject	LM
	EXP	0.03	0.39	0.02	Reject	LM
	G	0.05	0.40	0.01	Reject	LM
2016	RR	-0.41	-0.06	0.01	Reject	RR
	EXP	-0.44	-0.11	0.00	Reject	EXP
	G	-0.45	-0.12	0.00	Reject	G
2018	RR	-0.33	0.15	0.45	Accept	
	EXP	-0.32	0.14	0.44	Accept	
	G	-0.33	0.13	0.39	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

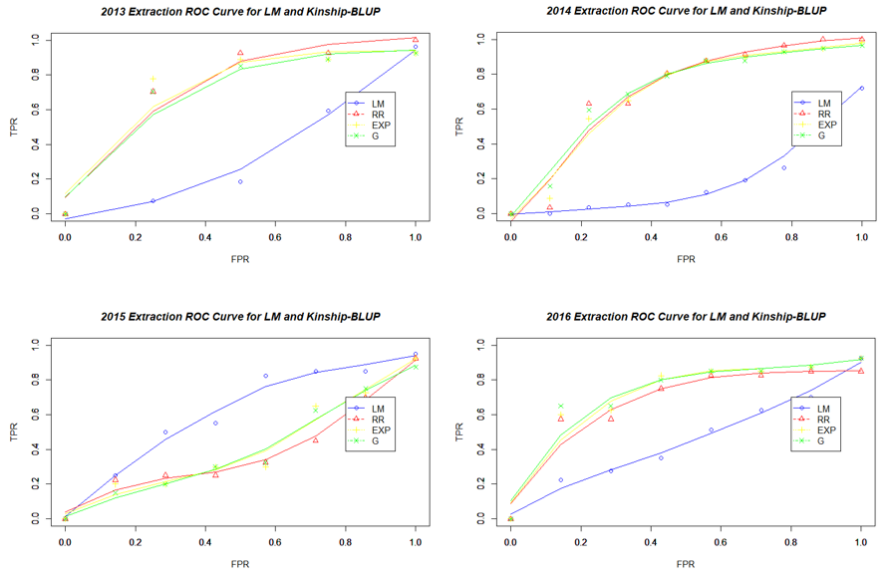

Figure 3. ROC curves of LM and G-BLUP for flour extraction in 2013, 2014, 2015 and 2016

Table 6 shows the two approaches are not significantly different in the years 2012, 2016 and 2018 for mixograph prediction. The ROC curves corresponding to the test with significant

differences are shown in Figure 4. G-BLUP performs better in 2015, and LM provides better results in 2013 and 2014. There is no clear pattern as to which method is always better than the other. But we can see that LM is not worse than G-BLUP.

Table 6. LM VS G-BLUP: P-values and 95% CIs for mixograph prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\hat{\theta}^s = \hat{\theta}^r$	BETTER METHOD
2012	RR	-0.36	-0.06	0.06	Accept	
	EXP	-0.42	-0.10	0.05	Accept	
	G	-0.44	-0.12	0.05	Accept	
2013	RR	0.01	0.39	0.04	Reject	LM
	EXP	0.00	0.35	0.05	Reject	LM
	G	0.01	0.36	0.04	Reject	LM
2014	RR	0.01	0.30	0.03	Reject	LM
	EXP	0.03	0.31	0.02	Reject	LM
	G	0.02	0.30	0.02	Reject	LM
2015	RR	-0.35	-0.05	0.01	Reject	RR
	EXP	-0.35	-0.04	0.01	Reject	EXP
	G	-0.33	-0.03	0.02	Reject	G
2016	RR	-0.16	0.20	0.85	Accept	
	EXP	-0.15	0.21	0.72	Accept	
	G	-0.16	0.20	0.82	Accept	
2018	RR	-0.41	0.09	0.20	Accept	
	EXP	-0.46	0.00	0.06	Accept	
	G	-0.44	0.04	0.11	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

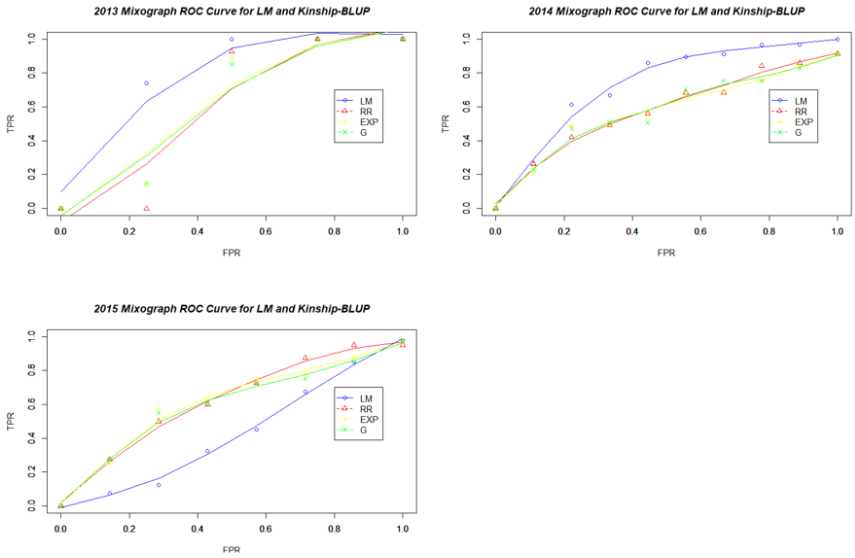

Figure 4. ROC curves of LM and G-BLUP for mixograph in 2013, 2014 and 2015

The resulting AUC analysis in Table 7 indicates there is no significant difference between the proposed method and G-BLUP, except for the year 2012. This shows that these methods provide almost the same performance for bake absorption prediction. Figure 5 provides the ROC curve corresponding to the test with significant differences is plotted.

Table 7. LM VS G-BLUP: P-values and 95% CIs for bake absorption prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.27	0.00	0.05	Reject	RR
	EXP	-0.28	-0.01	0.03	Reject	EXP
	G	-0.28	-0.01	0.03	Reject	G
2013	RR	-0.44	0.02	0.07	Accept	
	EXP	-0.37	0.09	0.24	Accept	
	G	-0.37	0.10	0.25	Accept	
2014	RR	-0.21	0.09	0.44	Accept	
	EXP	-0.19	0.12	0.64	Accept	
	G	-0.19	0.11	0.62	Accept	
2015	RR	-0.03	0.34	0.10	Accept	
	EXP	-0.03	0.34	0.09	Accept	
	G	-0.03	0.34	0.09	Accept	
2016	RR	-0.22	0.05	0.22	Accept	
	EXP	-0.24	0.05	0.19	Accept	
	G	-0.23	0.05	0.21	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

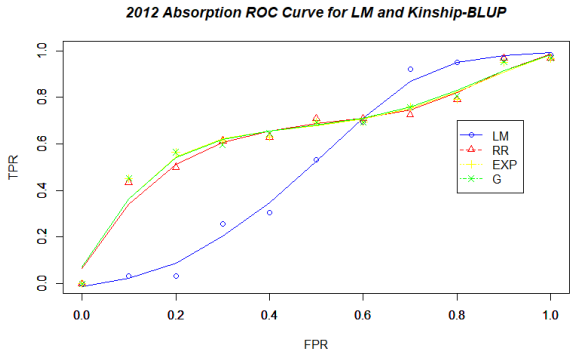

Figure 5. ROC curves of LM and G-BLUP for bake absorption in 2012

4.2. Bayesian Linear Model

Next, we compared the AUCs of the loaf volume prediction result of the Bayesian linear model and G-BLUP. Table 8 lists the resulting P-values and 95% confidence intervals. In 2012,

2013, 2014, and 2016, there were no significant differences between BAY and G-BLUP. In addition, BAY obtained more accurate forecasts in 2015 and 2018(Figure 6). In short, the performance of BAY is similar to or better than G-BLUP when predicting loaf volume.

Table 8. BAY VS G-BLUP: P-values and 95% CIs for loaf volume prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.22	0.02	0.09	Accept	
	EXP	-0.21	0.03	0.13	Accept	
	G	-0.21	0.03	0.13	Accept	
2013	RR	-0.37	0.03	0.10	Accept	
	EXP	-0.37	0.02	0.08	Accept	
	G	-0.37	0.03	0.10	Accept	
2014	RR	-0.12	0.09	0.77	Accept	
	EXP	-0.13	0.09	0.72	Accept	
	G	-0.13	0.08	0.58	Accept	
2015	RR	0.07	0.28	0.00	Reject	BAY
	EXP	0.05	0.27	0.01	Reject	BAY
	G	0.06	0.27	0.00	Reject	BAY
2016	RR	-0.20	0.04	0.18	Accept	
	EXP	-0.23	0.00	0.06	Accept	
	G	-0.22	0.02	0.09	Accept	
2018	RR	0.19	0.52	0.00	Reject	BAY
	EXP	0.18	0.51	0.00	Reject	BAY
	G	0.18	0.52	0.00	Reject	BAY

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

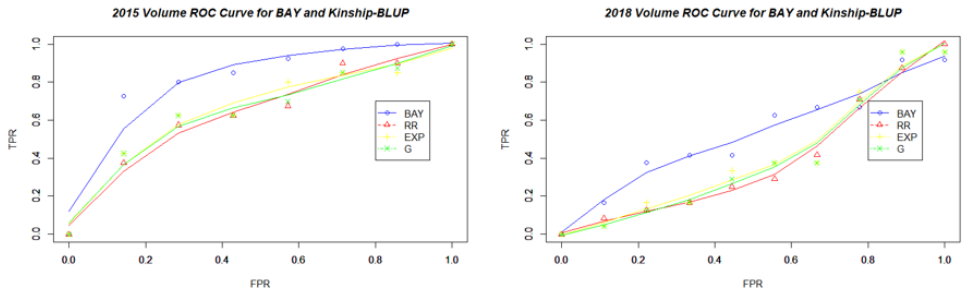

Figure 6. ROC curves of BAY and G-BLUP for loaf volume in 2015 and 2018

In Table 9, the AUC results are provided for prediction flour extraction. There were no significant differences between BAY and G-BLUP in 2012 and 2018, but G-BLUP provided better results based on its AUC in 2013, 2014 and 2016. Bay performs better in 2015. Figure 7 shows the ROC curves corresponding to tests with significant differences.

Table 9. BAY VS RR-BLUP: P-values and 95% CIs for flour extraction prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.23	0.08	0.35	Accept	
	EXP	-0.24	0.06	0.24	Accept	
	G	-0.24	0.07	0.28	Accept	
2013	RR	-0.75	-0.34	0.00	Reject	RR
	EXP	-0.75	-0.30	0.00	Reject	EXP
	G	-0.73	-0.27	0.00	Reject	G
2014	RR	-0.60	-0.26	0.00	Reject	RR
	EXP	-0.59	-0.25	0.00	Reject	EXP
	G	-0.60	-0.26	0.00	Reject	G
2015	RR	0.03	0.39	0.02	Reject	BAY
	EXP	0.01	0.36	0.04	Reject	BAY
	G	0.02	0.37	0.03	Reject	BAY
2016	RR	-0.40	-0.07	0.01	Reject	RR
	EXP	-0.44	-0.11	0.00	Reject	EXP
	G	-0.45	-0.12	0.00	Reject	G
2018	RR	-0.13	0.25	0.54	Accept	
	EXP	-0.14	0.23	0.65	Accept	
	G	-0.14	0.23	0.64	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

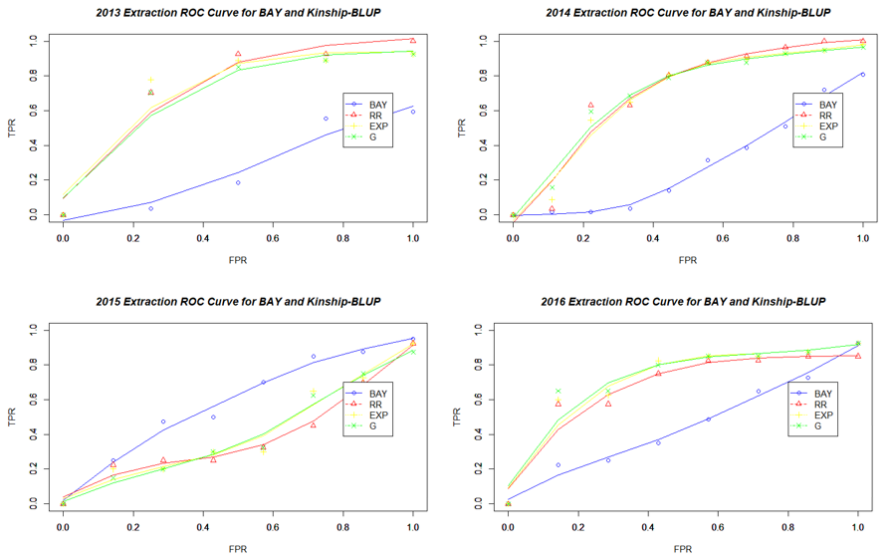

Figure 7. ROC curves of BAY and G-BLUP for flour extraction in 2013, 2014, 2015 and 2016

Table 10 shows that the two methods have no significant differences in the mixograph predictions in 2013, 2014, 2016, and 2018. Figure 8 indicates that G-BLUP performed better in 2012 and 2015. This shows that BAY is not a suitable method for predicting mixograph.

Table 10. BAY VS G-BLUP: P-values and 95% CIs for mixograph prediction

YEAR	KERNEL	CI		P-VALUE	$H0:\hat{\theta}^s = \hat{\theta}^r$	BETTER METHOD
2012	RR	-0.36	-0.06	0.01	Reject	RR
	EXP	-0.42	-0.10	0.00	Reject	EXP
	G	-0.44	-0.12	0.00	Reject	G
2013	RR	-0.02	0.37	0.08	Accept	
	EXP	-0.03	0.33	0.11	Accept	
	G	-0.02	0.34	0.09	Accept	
2014	RR	-0.02	0.27	0.09	Accept	
	EXP	-0.01	0.28	0.06	Accept	
	G	-0.01	0.27	0.07	Accept	
2015	RR	-0.36	-0.04	0.01	Reject	RR
	EXP	-0.36	-0.04	0.02	Reject	EXP
	G	-0.34	-0.03	0.02	Reject	G
2016	RR	-0.17	0.20	0.88	Accept	
	EXP	-0.16	0.21	0.76	Accept	
	G	-0.17	0.20	0.85	Accept	
2018	RR	-0.02	0.24	0.09	Accept	
	EXP	-0.02	0.24	0.11	Accept	
	G	-0.02	0.24	0.09	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

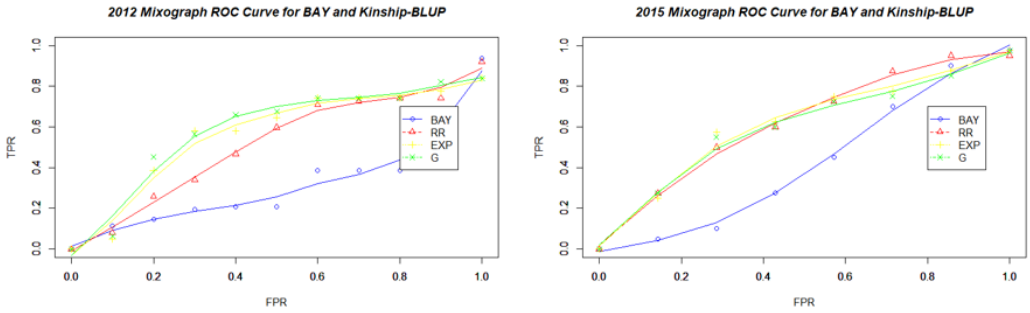

Figure 8. ROC curves of BAY and G-BLUP for mixograph in 2013 and 2015

In Figure 9, the ROC curve corresponding to the test with significant differences is plotted. The P-values and 95% confidence intervals in Table 11 indicate that, except for 2012, there is no significant difference between the BAY and G-BLUP. This shows that these methods have similar bake absorption prediction performance. However, based on the fact that the predictive ability of G-BLUP was higher than the proposed methods, G-BLUP provided marginally better performance.

Table 11. BAY VS RR-BLUP: P-values and 95% CIs for bake absorption prediction

YEAR	KERNEL	CI		P-VALUE	$H0:\hat{\theta}^s = \hat{\theta}^r$	BETTER METHOD
2012	RR	-0.27	0.00	0.05	Reject	RR
	EXP	-0.28	-0.01	0.03	Reject	EXP
	G	-0.28	-0.01	0.03	Reject	G
2013	RR	-0.44	0.02	0.07	Accept	
	EXP	-0.37	0.09	0.24	Accept	
	G	-0.37	0.10	0.25	Accept	
2014	RR	-0.21	0.09	0.44	Accept	
	EXP	-0.19	0.12	0.63	Accept	
	G	-0.19	0.11	0.62	Accept	
2015	RR	-0.03	0.34	0.10	Accept	
	EXP	-0.03	0.34	0.09	Accept	
	G	-0.03	0.34	0.09	Accept	
2016	RR	-0.22	0.05	0.22	Accept	
	EXP	-0.24	0.05	0.19	Accept	
	G	-0.23	0.05	0.21	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

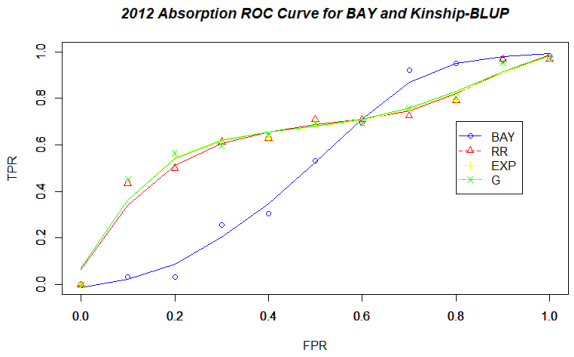

Figure 9. ROC curves of BAY and G-BLUP for bake absorption in 2012

4.3. K Nearest Neighbors

Using to the root mean square error (RMSE) results of cross-validation, we selected the k value for each response variable with K-NN as described in section 3.5. Also, we compared AUC of K-NN and G-BLUP with the same nonparametric approach as with the other proposed methods. Table 12 and Figure 10 reveal that in the years 2015 and 2018, K-NN provided better loaf volume classification results than G-BLUP. This non-parametric approach also provides similar or better performance in loaf volume prediction compared to G-BLUP.

Table 12. K-NN VS G-BLUP: P-values and 95% CIs for loaf volume prediction

YEAR	KERNEL	CI		P-VALUE	$H0:\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.23	0.00	0.05	Accept	
	EXP	-0.22	0.01	0.07	Accept	
	G	-0.22	0.01	0.07	Accept	
2013	RR	-0.33	0.06	0.16	Accept	
	EXP	-0.34	0.05	0.13	Accept	
	G	-0.33	0.06	0.16	Accept	
2014	RR	-0.14	0.08	0.58	Accept	
	EXP	-0.15	0.08	0.55	Accept	
	G	-0.15	0.06	0.43	Accept	
2015	RR	0.06	0.29	0.00	Reject	K-NN
	EXP	0.04	0.27	0.01	Reject	K-NN
	G	0.06	0.28	0.00	Reject	K-NN
2016	RR	-0.18	0.07	0.40	Accept	
	EXP	-0.21	0.03	0.15	Accept	
	G	-0.19	0.05	0.24	Accept	
2018	RR	0.14	0.48	0.00	Reject	K-NN
	EXP	0.13	0.48	0.00	Reject	K-NN
	G	0.14	0.49	0.00	Reject	K-NN

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

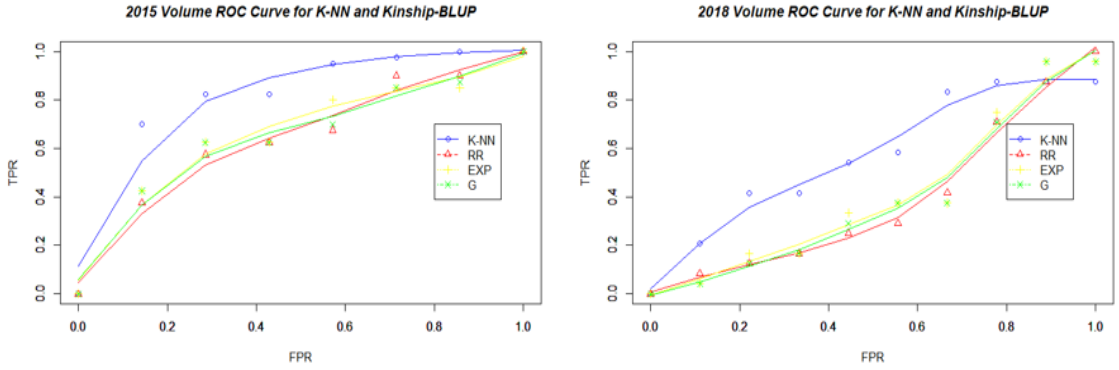

Figure 10. ROC curves of K-NN and G-BLUP for loaf volume in 2015 and 2018

Table 13 shows that based on the AUC P-values and 95% confidence interval, G-BLUP is better for predicting flour extraction than K-NN. The K-NN are similar to those of LM in that the predictor TW is not significant for the response flour extraction. The ROC curves corresponding to the test with significant differences are shown in Figure 11.

Table 13. K-NN VS RR-BLUP: P-values and 95% CIs for flour extraction prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.29	0.01	0.07	Accept	
	EXP	-0.30	-0.01	0.04	Reject	EXP
	G	-0.29	0.00	0.05	Reject	G
2013	RR	-0.67	-0.32	0.00	Reject	RR
	EXP	-0.67	-0.28	0.00	Reject	EXP
	G	-0.64	-0.25	0.00	Reject	G
2014	RR	-0.36	-0.03	0.02	Reject	RR
	EXP	-0.35	-0.02	0.03	Reject	EXP
	G	-0.35	-0.03	0.02	Reject	G
2015	RR	-0.25	0.12	0.49	Accept	
	EXP	-0.29	0.10	0.36	Accept	
	G	-0.27	0.11	0.41	Accept	
2016	RR	-0.24	0.10	0.43	Accept	
	EXP	-0.28	0.06	0.19	Accept	
	G	-0.28	0.05	0.16	Accept	
2018	RR	-0.24	0.13	0.55	Accept	
	EXP	-0.25	0.11	0.43	Accept	
	G	-0.25	0.11	0.43	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

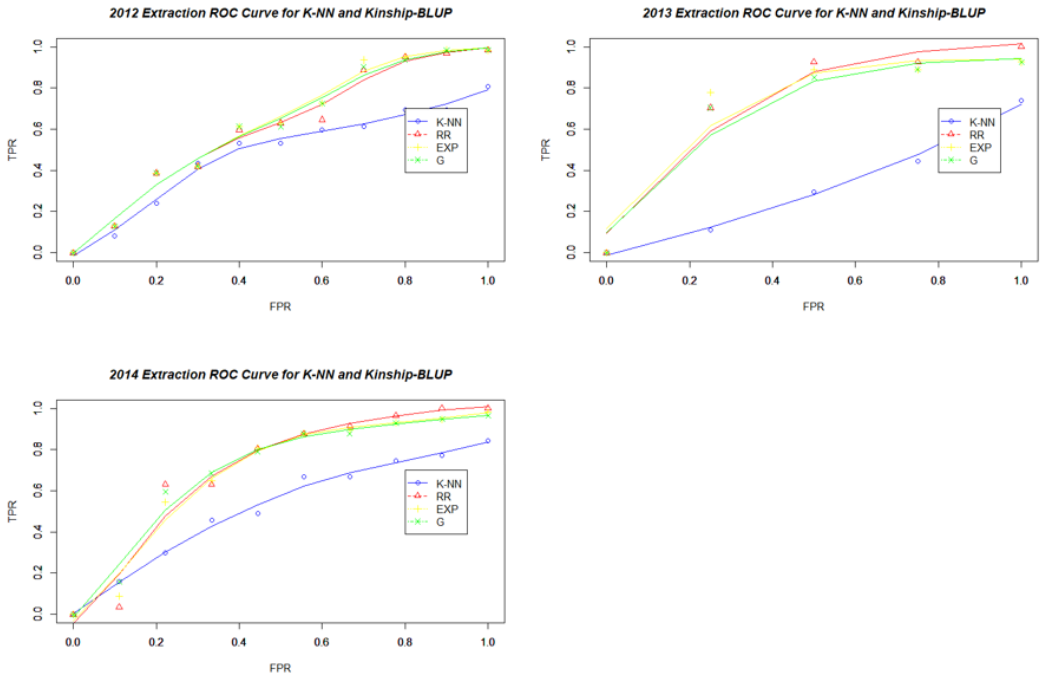

Figure 11. ROC curves of K-NN and G-BLUP for flour extraction in 2012, 2013 and 2014

It can be seen from Table 14 that K-NN is similar to or better than G-BLUP in predicting mixograph. These results shows that the two methods have no significant differences in the mixograph predictions in 2012, 2016, and 2018. Figure 12 shows the ROC curves in 2013 and 2014, when K-NN performs better than G-BLUP.

Table 14. K-NN VS G-BLUP: P-values and 95% CIs for mixograph prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.20	0.08	0.41	Accept	
	EXP	-0.25	0.03	0.13	Accept	
	G	-0.27	0.01	0.06	Accept	
2013	RR	0.04	0.39	0.02	Reject	K-NN
	EXP	0.03	0.34	0.02	Reject	K-NN
	G	0.04	0.35	0.02	Reject	K-NN
2014	RR	0.03	0.30	0.02	Reject	K-NN
	EXP	0.04	0.31	0.01	Reject	K-NN
	G	0.03	0.30	0.01	Reject	K-NN
2015	RR	-0.29	0.05	0.17	Accept	
	EXP	-0.28	0.06	0.20	Accept	
	G	-0.27	0.07	0.26	Accept	
2016	RR	-0.24	0.07	0.27	Accept	
	EXP	-0.23	0.08	0.37	Accept	
	G	-0.24	0.07	0.30	Accept	
2018	RR	-0.12	0.18	0.67	Accept	
	EXP	-0.12	0.18	0.70	Accept	
	G	-0.11	0.18	0.66	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

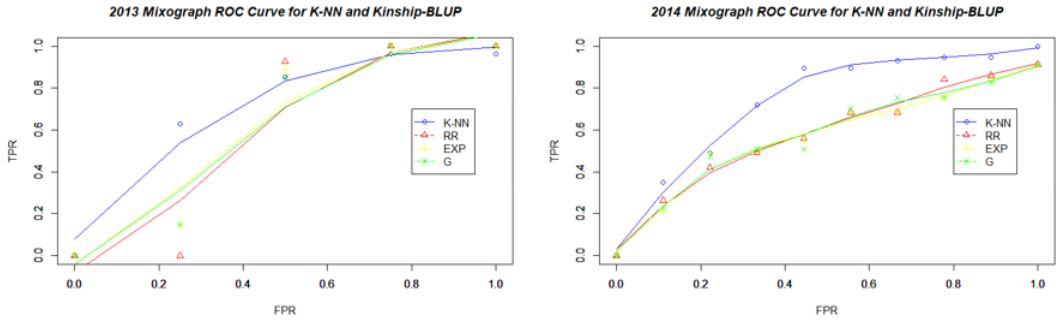

Figure 12. ROC curves of K-NN and G-BLUP for mixograph in 2012 and 2014

The AUC P-values and 95% confidence intervals obtained in Table 15 indicate that, except for 2012, there is no significant difference between K-NN and G-BLUP. This shows that

these two methods can provide similar bake absorption prediction performance. In Figure 13, the ROC curve corresponding to the test with significant differences is plotted.

Table 15. K-NN VS G-BLUP: P-values and 95% CIs for bake absorption prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.35	-0.09	0.00	Reject	RR
	EXP	-0.35	-0.10	0.00	Reject	EXP
	G	-0.35	-0.10	0.00	Reject	G
2013	RR	-0.37	0.06	0.16	Accept	
	EXP	-0.29	0.14	0.47	Accept	
	G	-0.30	0.14	0.48	Accept	
2014	RR	-0.21	0.02	0.11	Accept	
	EXP	-0.19	0.05	0.24	Accept	
	G	-0.19	0.05	0.23	Accept	
2015	RR	-0.13	0.24	0.55	Accept	
	EXP	-0.12	0.24	0.53	Accept	
	G	-0.12	0.24	0.53	Accept	
2016	RR	-0.22	0.04	0.18	Accept	
	EXP	-0.23	0.03	0.14	Accept	
	G	-0.22	0.04	0.17	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

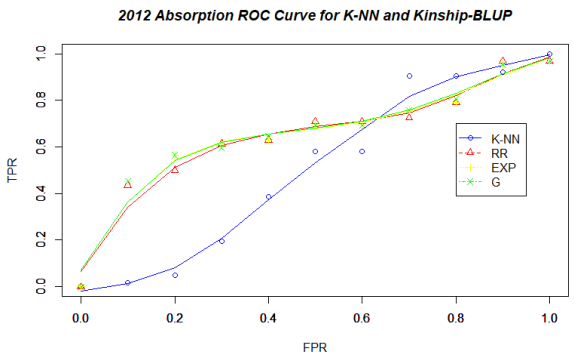

Figure 13. ROC curves of K-NN and G-BLUP for bake absorption in 2012

4.4. Random Forest

We compared the AUC of Random forest and G-BLUP with the Mann-Whitney two-sample test. Table 16 shows that in 2015 and 2018, RF can provide better loaf volume prediction results than G-BLUP. In Figure 14, the ROC curves corresponding to the tests with significant

differences is shown. This non-linear model also provides better performance in loaf volume prediction.

Table 16. RF VS G-BLUP: P-values and 95% CIs for loaf volume prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.19	0.03	0.16	Accept	
	EXP	-0.18	0.04	0.21	Accept	
	G	-0.18	0.04	0.21	Accept	
2013	RR	-0.38	0.01	0.06	Accept	
	EXP	-0.39	0.00	0.05	Accept	
	G	-0.38	0.01	0.06	Accept	
2014	RR	-0.13	0.09	0.70	Accept	
	EXP	-0.13	0.09	0.66	Accept	
	G	-0.14	0.07	0.53	Accept	
2015	RR	0.07	0.28	0.00	Reject	RF
	EXP	0.05	0.27	0.01	Reject	RF
	G	0.06	0.27	0.00	Reject	RF
2016	RR	-0.18	0.06	0.33	Accept	
	EXP	-0.21	0.02	0.11	Accept	
	G	-0.19	0.04	0.19	Accept	
2018	RR	0.14	0.49	0.00	Reject	RF
	EXP	0.12	0.48	0.00	Reject	RF
	G	0.13	0.49	0.00	Reject	RF

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

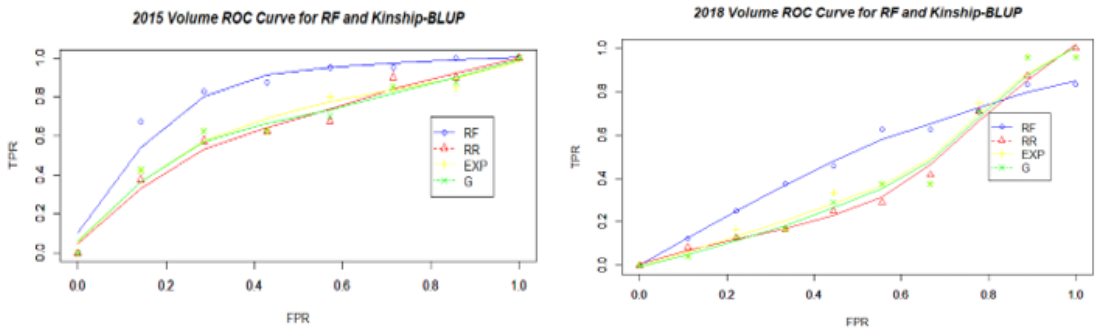

Figure 14. ROC curves of RF and G-BLUP for loaf volume in 2018

Table 17 shows that, based on the AUC P-values and 95% confidence intervals, G-BLUP is a better method of predicting flour extraction than RF in the years 2013, 2014, and 2016 (Figure 15). There is no significant difference between the two methods in 2013, 2015, and 2018.

Table 17. RF VS G-BLUP: P-values and 95% CIs for flour extraction prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\widehat{\theta}^s = \widehat{\theta}^r$	BETTER METHOD
2012	RR	-0.27	0.04	0.15	Accept	
	EXP	-0.28	0.02	0.09	Accept	
	G	-0.28	0.03	0.11	Accept	
2013	RR	-0.55	-0.20	0.00	Reject	RR
	EXP	-0.56	-0.16	0.00	Reject	EXP
	G	-0.53	-0.13	0.00	Reject	G
2014	RR	-0.30	-0.09	0.00	Reject	RR
	EXP	-0.28	-0.07	0.00	Reject	EXP
	G	-0.29	-0.08	0.00	Reject	G
2015	RR	-0.22	0.15	0.71	Accept	
	EXP	-0.25	0.13	0.53	Accept	
	G	-0.24	0.14	0.60	Accept	
2016	RR	-0.41	-0.10	0.00	Reject	RR
	EXP	-0.45	-0.15	0.00	Reject	EXP
	G	-0.46	-0.15	0.00	Reject	G
2018	RR	-0.01	0.36	0.06	Accept	
	EXP	-0.02	0.33	0.09	Accept	
	G	-0.02	0.33	0.08	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

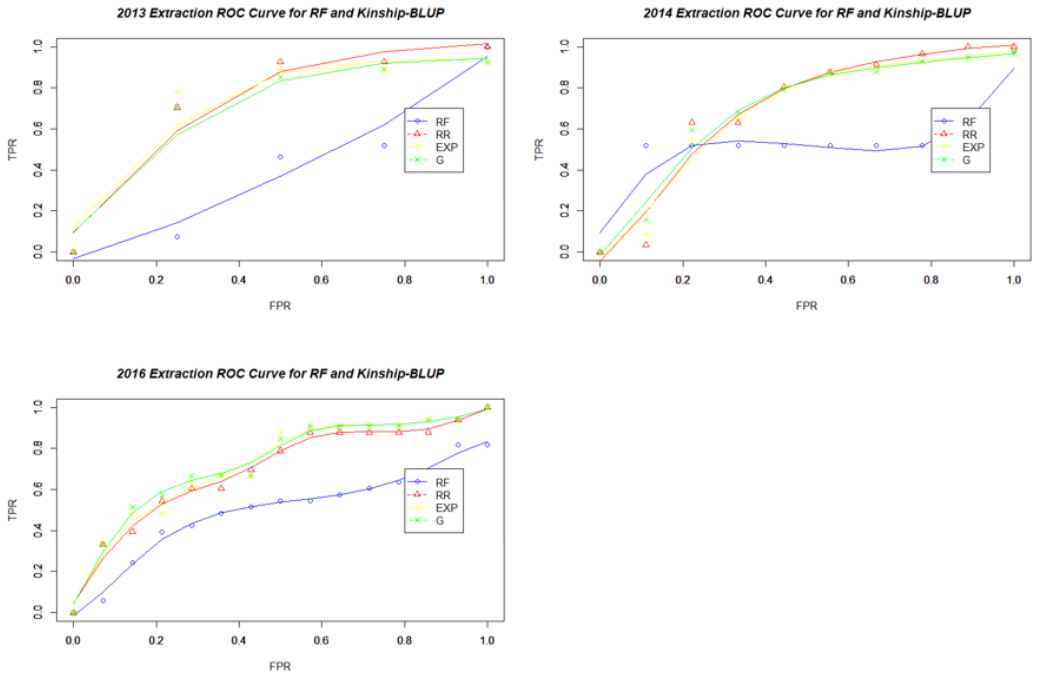

Figure 15. ROC curves of RF and G-BLUP for flour extraction in 2013, 2014 and 2016

Table 18 shows that the two methods have no significant differences in the mixograph predictions in most years. G-BLUP performs better only in 2015 (Figure 15). These two methods provide similar mixograph prediction performance.

Table 18. RF VS G-BLUP: P-values and 95% CIs for mixograph prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\hat{\theta}^s = \hat{\theta}^r$	BETTER METHOD
2012	RR	-0.20	0.08	0.42	Accept	
	EXP	-0.26	0.04	0.16	Accept	
	G	-0.29	0.02	0.10	Accept	
2013	RR	-0.12	0.31	0.40	Accept	
	EXP	-0.14	0.27	0.53	Accept	
	G	-0.13	0.28	0.47	Accept	
2014	RR	-0.02	0.27	0.09	Accept	
	EXP	0.00	0.28	0.06	Accept	
	G	-0.01	0.27	0.07	Accept	
2015	RR	-0.42	-0.07	0.01	Reject	RR
	EXP	-0.41	-0.07	0.01	Reject	EXP
	G	-0.40	-0.05	0.01	Reject	G
2016	RR	-0.33	0.08	0.24	Accept	
	EXP	-0.31	0.09	0.29	Accept	
	G	-0.32	0.08	0.25	Accept	
2018	RR	-0.09	0.21	0.41	Accept	
	EXP	-0.09	0.21	0.44	Accept	
	G	-0.08	0.21	0.40	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

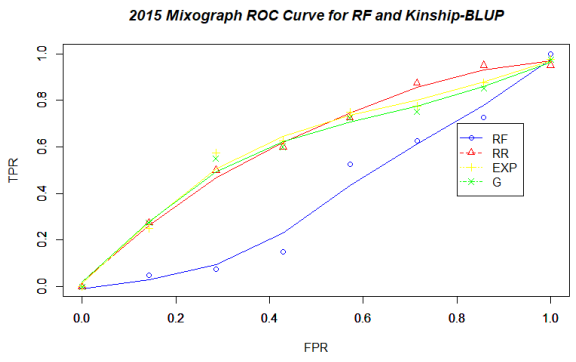

Figure 16. ROC curves of RF and G-BLUP for mixograph in 2015

Table 19 shows the two approaches are not significantly different in the year 2013, 2014 and 2016 for bake absorption prediction. The G-BLUP performs better in 2012, and RF provides

better results in 2015. In Figure 17, the ROC curves corresponding to the tests with significant differences are plotted. There is no clear pattern which method is always better than the other. But we can see that our approach is not worse than G-BLUP.

Table 19. RF VS RR-BLUP: P-values and 95% CIs for bake absorption prediction

YEAR	KERNEL	CI		P-VALUE	H0: $\hat{\theta}^s = \hat{\theta}^r$	BETTER METHOD
2012	RR	-0.31	-0.05	0.01	Reject	RR
	EXP	-0.32	-0.05	0.01	Reject	EXP
	G	-0.32	-0.06	0.01	Reject	G
2013	RR	-0.41	0.02	0.08	Accept	
	EXP	-0.34	0.10	0.28	Accept	
	G	-0.34	0.10	0.29	Accept	
2014	RR	-0.28	0.03	0.11	Accept	
	EXP	-0.26	0.05	0.20	Accept	
	G	-0.26	0.05	0.19	Accept	
2015	RR	0.06	0.39	0.01	Reject	RF
	EXP	0.07	0.39	0.01	Reject	RF
	G	0.07	0.40	0.01	Reject	RF
2016	RR	-0.26	0.39	0.06	Accept	
	EXP	-0.27	0.39	0.11	Accept	
	G	-0.27	0.40	0.20	Accept	

Note: RR = ridge regression; EXP = exponential; G = Gaussian.

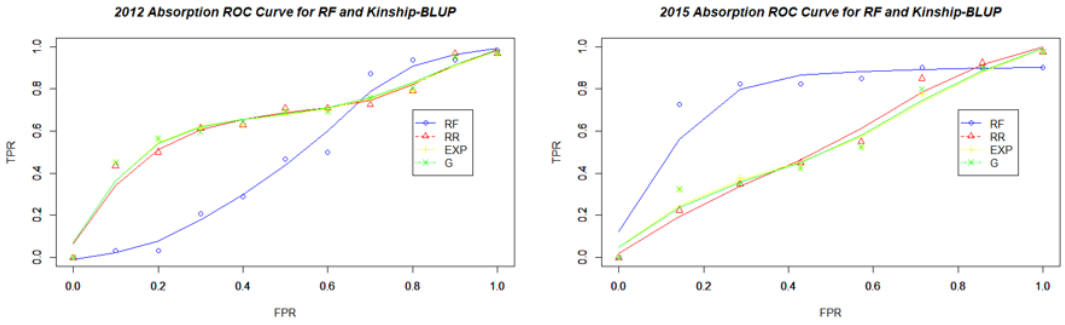

Figure 17. ROC curves of RF and G-BLUP for bake absorption in 2012 and 2015

4.5. Original Data Set Results and Discussion

We compared a total of seven prediction models, including linear and nonlinear (Table 20). The three G-BLUP methods almost provide the same results in the prediction of loaf volume, flour extraction, mixograph classification, and bake absorption. Based on the results obtained above, we conclude that the performance of the models without marker information is

similar to or better than the models with marker information for loaf volume prediction. However, for flour extraction prediction, the G-BLUP methods provide similar or better results than the proposed methods. For mixograph classification prediction, when comparing the linear model and the G-BLUP methods, there is no clear pattern as to which method is always better. The results of G-BLUP are similar to or better than the Bayesian linear model. K-nearest neighbors provides similar or better performance than G-BLUP. Random forest and G-BLUP methods can provide almost the same performance of mixograph classification prediction. Linear model, Bayesian linear model and k-nearest neighbors can provide almost the same bake absorption prediction performance as G-BLUP. Although there is no clear pattern indicating which method between random forest and G-BLUP is always better, we can see that the proposed approach is not worse than G-BLUP.

Table 20. Seven prediction models

MODELS WITH MARKER INFORMATION	**MODELS WITHOUT MARKER INFORMATION**
G-BLUP(RR-BLUP)	Linear model
G-BLUP (Gaussian)	Bayesian linear model
G-BLUP (Exponential)	K-NN
	Random Forest

4.5.1. Comparison of proposed method using original data set

Up to now, each of the proposed methods were compared to G-BLUP. Now, we compare the proposed methods to each other to find the best method of loaf volume prediction using the non-parametric Mann-Whitney two-sample statistic to compare AUCs. We chose loaf volume because the proposed methods provided a better performance than predicting flour extraction, bake absorption and mixograph. From the results of the 95% confidence intervals and p-values in Table 21, it can be found that there is no significant difference between these methods.

Table 21. LM VS K-NN VS RF VS BAY: P-values and 95% CIs for loaf volume prediction

		LM			K-NN			RF			BAY		
		CI		P-value	CI		P-value	CI		P-value	CI		P-value
2012	LM	-	-	-	-0.05	0.01	0.27	-0.04	0.00	0.11	0.00	0.00	1.00
	K-NN	-0.05	0.01	0.27	-	-	-	-0.07	0.00	0.06	-0.04	0.00	0.11
	RF	-0.04	0.00	0.11	-0.07	0.00	0.06	-	-	-	-0.01	0.05	0.27
	BAY	0.00	0.00	1.00	-0.04	0.00	0.11	-0.01	0.05	0.27	-	-	-
2013	LM	-	-	-	-0.01	0.07	0.19	-0.03	0.06	0.43	0.00	0.00	1.00
	K-NN	-0.01	0.07	0.19	-	-	-	-0.02	0.11	0.15	-0.03	0.06	0.43
	RF	-0.03	0.06	0.43	-0.02	0.11	0.15	-	-	-	-0.07	0.01	0.19
	BAY	0.00	0.00	1.00	-0.03	0.06	0.43	-0.07	0.01	0.19	-	-	-
2014	LM	-	-	-	-0.03	0.00	0.14	-0.01	0.02	0.53	-0.01	0.01	1.00
	K-NN	-0.03	0.00	0.14	-	-	-	-0.03	0.01	0.30	-0.01	0.02	0.59
	RF	-0.01	0.02	0.53	-0.03	0.01	0.30	-	-	-	0.00	0.03	0.13
	BAY	0.00	0.00	1.00	-0.01	0.02	0.59	0.00	0.03	0.13	-	-	-
2015	LM	-	-	-	-0.02	0.02	1.00	-0.02	0.02	1.00	0.00	0.00	1.00
	K-NN	-0.02	0.02	1.00	-	-	-	-0.02	0.02	1.00	-0.02	0.02	1.00
	RF	-0.02	0.02	1.00	-0.02	0.02	1.00	-	-	-	-0.02	0.02	1.00
	BAY	0.00	0.00	1.00	-0.02	0.02	1.00	-0.02	0.02	1.00	-	-	-
2016	LM	-	-	-	0.00	0.06	0.07	-0.06	0.01	0.22	0.00	0.00	1.00
	K-NN	0.00	0.06	0.07	-	-	-	-0.03	0.05	0.79	-0.06	0.01	0.22
	RF	-0.06	0.01	0.22	-0.03	0.05	0.79	-	-	-	-0.06	0.00	0.07
	BAY	0.00	0.00	1.00	-0.06	0.01	0.22	-0.06	0.00	0.07	-	-	-
2018	LM	-	-	-	-0.15	0.03	0.17	-0.02	0.12	0.15	0.00	0.00	1.00
	K-NN	-0.15	0.03	0.17	-	-	-	-0.10	0.08	0.83	-0.02	0.12	0.15
	RF	-0.02	0.12	0.15	-0.10	0.08	0.83	-	-	-	-0.03	0.15	0.17
	BAY	0.00	0.00	1.00	-0.02	0.12	0.15	-0.03	0.15	0.17	-	-	-

Note: BAY = Bayesian linear model; LM = linear model; K-NN = K nearest neighbor; RF = random forest.

4.5.2. PPV and TPR of original data set

In addition to predictive ability, we also include the results in Table 22 to show positive predictive value (PPV) and sensitivity (TPR). PPV is the proportion of lines identified as underperforming that are actually in the underperforming group. TPR is the proportion of underperforming lines identified as underperforming.

In 2012, our proposed methods have higher TPR and PPV than G-BLUP in terms of mixograph and bake absorption prediction. In 2013, the proposed methods achieve similar or

better performance in mixograph and bake absorption. G-BLUP performs better in 2014 for flour extraction and bake absorption. In 2015 the proposed methods have the same or better TPR and PPV for all response predictions. In the 2016 flour extraction and mixograph analysis, the proposed methods are not worse than G-BLUP. In 2018, for all response predictions, proposed methods have the same or better TPR and PPV. These results indicate that LM, K-NN and BAY are not worse than G-BLUP methods in prediction of loaf volume, mixograph and bake absorption.

Table 22. PPV and TPR of seven models for 4 response variables

YEAR	METHOD	VOLUME		EXTRACTION		MIXOGRAPH		ABSORPTION	
		TPR	PPV	TPR	PPV	TPR	PPV	TPR	PPV
2012	LM	0.40	0.40	0.10	0.10	0.10	0.17	0.40	0.40
	BAY	0.40	0.40	0.10	0.10	0.10	0.17	0.40	0.40
	K-NN	0.40	0.40	0.10	0.10	0.10	0.17	0.40	0.40
	RF	0.40	0.40	0.10	0.10	0.14	0.14	0.30	0.30
	RR	0.30	0.30	0.30	0.30	0.10	0.17	0.20	0.20
	EXP	0.50	0.50	0.40	0.40	0.00	0.00	0.20	0.20
	G	0.50	0.50	0.40	0.40	0.00	0.00	0.20	0.20
2013	LM	0.25	0.25	0.25	0.25	0.75	0.75	0.25	0.25
	BAY	0.25	0.25	0.10	0.10	0.50	0.50	0.25	0.25
	K-NN	0.25	0.25	0.10	0.10	0.50	0.50	0.25	0.25
	RF	0.25	0.25	0.25	0.25	0.50	0.50	0.25	0.25
	RR	0.50	0.50	0.30	0.30	0.50	0.50	0.25	0.25
	EXP	0.50	0.50	0.25	0.25	0.50	0.50	0.00	0.00
	G	0.50	0.50	0.25	0.25	0.50	0.50	0.00	0.00
2014	LM	0.22	0.22	0.00	0.00	0.44	0.50	0.11	0.13
	BAY	0.22	0.22	0.10	0.10	0.33	0.38	0.25	0.25
	K-NN	0.22	0.22	0.10	0.10	0.44	0.50	0.11	0.13
	RF	0.22	0.22	0.33	0.11	0.33	0.38	0.11	0.13
	RR	0.22	0.22	0.44	0.44	0.11	0.13	0.33	0.38
	EXP	0.22	0.22	0.33	0.33	0.11	0.13	0.33	0.38
	G	0.22	0.22	0.33	0.33	0.11	0.13	0.33	0.38
2015	LM	0.57	0.57	0.14	0.14	0.14	0.14	0.14	0.14
	BAY	0.57	0.57	0.29	0.29	0.29	0.29	0.14	0.14
	K-NN	0.57	0.57	0.10	0.10	0.14	0.14	0.43	0.43
	RF	0.57	0.57	0.14	0.14	0.14	0.14	0.43	0.43
	RR	0.43	0.43	0.14	0.14	0.29	0.29	0.29	0.29
	EXP	0.14	0.14	0.14	0.14	0.29	0.29	0.29	0.29
	G	0.29	0.29	0.14	0.14	0.14	0.14	0.29	0.29
2016	LM	0.14	0.17	0.14	0.14	0.57	0.44	0.29	0.22
	BAY	0.14	0.17	0.14	0.14	0.43	0.33	0.29	0.22
	K-NN	0.14	0.17	0.29	0.29	0.43	0.33	0.14	0.11
	RF	0.14	0.17	0.14	0.14	0.57	0.44	0.13	0.11
	RR	0.29	0.33	0.14	0.14	0.29	0.22	0.43	0.33
	EXP	0.43	0.50	0.29	0.29	0.29	0.22	0.43	0.33
	G	0.43	0.50	0.29	0.29	0.29	0.22	0.43	0.33
2018	LM	0.14	0.14	0.29	0.29	0.10	0.10	0.14	0.17
	BAY	0.14	0.14	0.14	0.14	0.29	0.29	0.14	0.17
	K-NN	0.14	0.14	0.10	0.10	0.14	0.14	0.14	0.11
	RF	0.00	0.00	0.14	0.14	0.29	0.29	0.00	0.00
	RR	0.14	0.14	0.14	0.14	0.29	0.29	0.14	0.17
	EXP	0.14	0.14	0.14	0.14	0.29	0.29	0.00	0.00
	G	0.14	0.14	0.14	0.14	0.29	0.29	0.14	0.17

Note: BAY = Bayesian linear model; LM = linear model; K-NN = K nearest neighbor; RF = random forest; RR = ridge regression; EXP = exponential; G = Gaussian.

4.6. New Data Set Results and Discussions

A potential reason why the predictive ability of our proposed methods is not as good as expected is that the number of independent predictors is too small and are not highly associated with the response variables. Therefore, we applied the proposed methods to a new data set that contains more predictors. There are two years in the new data set, 2018 and 2019, so we set the 2018 data as the training set and the 2019 data as the test set.

Table 23. Predictive abilities of LM, BAY, K-NN and RF for FABS, PKT, STAB and VOL

	FABS	PKT	STAB	VOL
LM	0.87	0.46	0.33	0.67
BAY	0.87	0.46	0.33	0.67
K-NN	0.69	0.32	0.29	0.40
RF	0.86	0.43	0.32	0.63

Note: BAY = Bayesian linear model; LM = linear model; K-NN = K nearest neighbor; RF = random forest; FABS = farinograph water absorption; PKT = farinograph peak time; STAB = farinograph stability; VOL = loaf volume.

There are 10 predictors (PMT, BEM, AM, PM, AE, SUE, PE, GPI and SRC) and 4 response variables (FABS, PKT, STAB, VOL). Predictive ability is shown in Table 23 Predictive abilities for the four traits were evaluated with proposed methods. The four proposed models conferred similar predictive abilities. The highest predictive ability approached 0.87 for farinograph water absorption, 0.46 for farinograph peak time, 0.33 for farinograph stability, and 0.67 for loaf volume. Since this data set has more predictors, it has higher accuracy, especially for linear model, Bayesian linear model and random forest methods.

To compare the performance of linear model, Bayesian linear model, k-nearest neighbor, and random forest on FABS, PKT, STAB and VOL quality traits, we also drew the ROC curves in Figure 18 and performed the Mann-Whitey method to test if the areas under the ROC curves are equivalent in Table 24. From the ROC curves and test results, the AUC of K-NN is smaller than the other three methods, especially for PKT and VOL prediction.

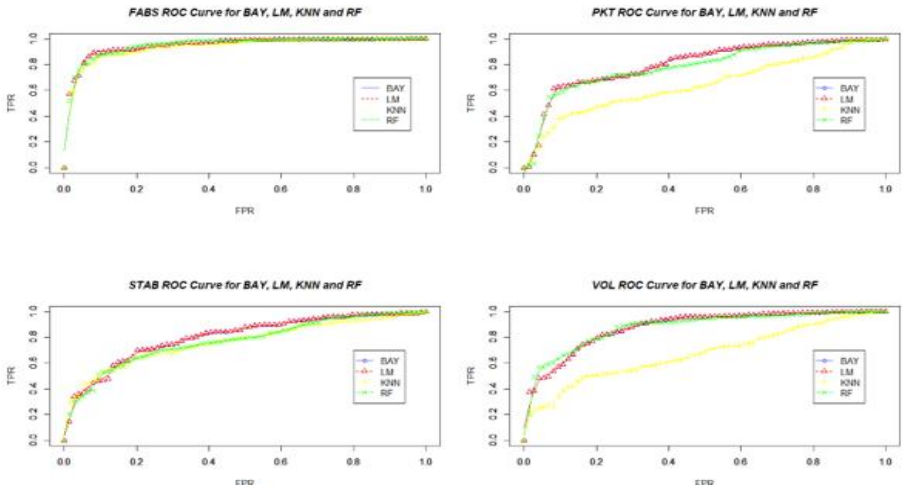

Figure 18. ROC curves of FABS, PKT, STAB and VOL using proposed methods

Table 24. P-values and 95% CIs for FABS, PKT, STAB and VOL using proposed methods

		LM			**K-NN**			**RF**			**BAY**		
		CI		P-value	CI		P-value	CI		P-value	CI		P-value
FABS	LM	-	-	-	**-0.02**	**0.00**	**0.02**	-0.01	0.01	0.82	0.00	0.00	1.00
	K-NN	**-0.02**	**0.00**	**0.02**	-	-	-	**0.00**	**0.02**	**0.02**	-0.01	0.01	0.82
	RF	-0.01	0.01	0.82	**0.00**	**0.02**	**0.01**	-	-	-	0.00	0.02	0.05
	BAY	0.00	0.00	1.00	-0.01	0.01	0.82	0.00	0.02	0.05	-	-	-
PKT	LM	-	-	-	-0.20	-0.13	0.00	-0.04	0.00	0.09	0.00	0.00	1.00
	K-NN	**-0.20**	**-0.13**	**0.00**	-	-	-	**0.11**	**0.18**	**0.00**	0.00	0.04	0.09
	RF	-0.04	0.00	0.09	**0.11**	**0.18**	**0.00**	-	-	-	0.13	0.20	0.05
	BAY	0.00	0.00	1.00	0.00	0.04	0.09	0.13	0.20	0.05	-	-	-
STAB	LM	-	-	-	**-0.06**	**-0.01**	**0.00**	**-0.05**	**-0.01**	**0.01**	0.00	0.00	1.00
	K-NN	**-0.06**	**-0.01**	**0.00**	-	-	-	-0.02	0.03	0.65	**0.01**	**0.05**	**0.01**
	RF	**-0.05**	**-0.01**	**0.01**	-0.02	0.03	0.65	-	-	-	**0.01**	**0.06**	**0.01**
	BAY	0.00	0.00	1.00	**0.01**	**0.05**	**0.01**	**0.01**	**0.06**	**0.01**	-	-	-
VOL	LM	-	-	-	-0.24	-0.17	0.00	-0.01	0.02	0.98	0.00	0.00	1.00
	K-NN	**-0.24**	**-0.17**	**0.00**	-	-	-	**0.17**	**0.24**	**0.00**	-0.02	0.01	1.00
	RF	-0.01	0.02	0.98	**0.17**	**0.24**	**0.00**	-	-	-	-0.07	0.04	0.05
	BAY	0.00	0.00	1.00	-0.02	0.01	0.98	-0.07	0.04	0.05	-	-	-

Note: BAY = Bayesian linear model; LM = linear model; K-NN = K nearest neighbor; RF = random forest; FABS = farinograph water absorption; PKT = farinograph peak time; STAB = farinograph stability; VOL = loaf volume.

Table 25 shows the TPR and PPV of proposed methods to predict FABS, PKT, STAB and VOL in the new dataset. The highest TPR/PPV approached 0.76/0.77 for farinograph water absorption (RF), 0.45/0.47 for farinograph peak time (LM and BAY), 0.43/0.42 for farinograph

stability (LM and BAY), and 0.62/0.64 for loaf volume (LM and BAY). This TPR/PPV analysis indicates that the proposed methods worked well, especially for FABS and VOL.

Table 25. TPR and PPV of proposed models in predicting FABS, PKT, STAB and VOL

	FABS		**PKT**		**STAB**		**VOL**	
	TPR	PPV	TPR	PPV	TPR	PPV	TPR	PPV
LM	0.72	0.73	0.45	0.47	0.43	0.42	0.62	0.64
BAY	0.72	0.73	0.45	0.47	0.43	0.42	0.62	0.64
K-NN	0.69	0.70	0.19	0.20	0.38	0.37	0.25	0.26
RF	0.76	0.77	0.39	0.41	0.38	0.37	0.57	0.58

Note: BAY = Bayesian linear model; LM = linear model; K-NN = K nearest neighbor; RF = random forest; FABS = farinograph water absorption; PKT = farinograph peak time; STAB = farinograph stability; VOL = loaf volume.

5. SUMMARY

In this paper, we developed four models without marker information for predicting wheat quality traits, including linear model, Bayesian linear model, k-nearest neighbors, and random forest. The goal of these proposed methods was to provide relatively accurate estimates of quality traits without the need for time and energy consumption for marker analysis. Then we used a non-parametric approach to infer whether the areas under the ROC curves of these models are equal. The better methods with significant difference were shown in Table 26.

Table 26. Better methods with significant difference

	VOLUME	EXTRACTION	MIXOGRAPH	ABSORPTION
2012		G-BLUP	G-BLUP	G-BLUP
2013	LM, BLM, K-NN, RF	G-BLUP	LM, BLM, K-NN	
2014		G-BLUP	LM, BLM, K-NN	
2015		LM, BLM	G-BLUP	RF
2016		G-BLUP		
2018	LM, BLM, K-NN, RF			

Note: BAY = Bayesian linear model; LM = linear model; K-NN = K nearest neighbor; RF = random forest

We successfully applied the proposed methods to the real spring wheat dataset and found that based on the results of Mann-Whitney tests and traditional predictive ability (Pearson correlation), all four proposed models performed better for predicting loaf volume than G-BLUP methods with marker information. The proposed methods are not suitable for flour extraction and bake absorption prediction, where the G-BLUP methods performed better. For mixograph classification, the proposed methods were not worse than G-BLUP, but it does not mean that we should utilize the prediction model. The TPR/PPV analysis indicates that LM, K-NN and BAY are not worse than G-BLUP methods in prediction of loaf volume, mixograph and bake absorption. In conclusion, the proposed methods with good effect and low cost can be used in combination with genome selection when predicting quality traits. But there were not consistent

results through years, further investigation is needed, and it is necessary to collect more breeding data.

We also applied the proposed methods to another wheat dataset and found it has good traditional predictive ability in predicting farinograph water absorption, farinograph peak time, and loaf volume, especially for linear model, Bayesian linear model and random forest methods. The TPR/PPV analysis performed on the new dataset indicates that the proposed methods performed better compared with the original dataset. Therefore, the proposed methods can provide better results when the dataset has more related predictors.

Future investigations are necessary to validate the kinds of conclusions that can be drawn from this study. There are two factors harming the accuracy of our proposed method, one is the number of predictors, another is the number of experimental data. The development of multi-trait/hybrid analysis to include both genotypic and new phenotypic chemical measurements will help to obtain further conclusions for this research. Increasing the size of the data set used for training to develop models will improve the predictive ability of this study. Weighting or selecting predictors based on the relationship between the predictors and response variables may produce better predictive models.

REFERENCES

Altman, Naomi S. 1992. An introduction to kernel and nearest-neighbor nonparametric regression. The American Statistician. 46 (3): 175–185. doi:10.1080/00031305.1992.10475879. hdl:1813/31637.

American Association of Cereal Chemists Approved Methods of Analysis, 11th Ed. AACC Method 54-40.02

Asoro, F.G., M.A. Newell, W.D. Beavis, M.P. Scott, and J.-L. Jannink. 2011. Accuracy and training population design for genomic selection on quantitative traits in elite North American oats. Plant Genome 4(2):132–144. doi:10.3835/plantgenome2011.02.0007.

Battenfield, Sarah D., et al. 2009. Genomic Selection for Processing and End-Use Quality Traits in the CIMMYT Spring Bread Wheat Breeding Program. Plant Genome doi:10.3835/plantgenome2016.01.0005.

Bernardo, R., and J. Yu. 2007. Prospects for genomewide selection for quantitative traits in maize. Crop Sci. 47:1082–1090.

Box, G. E. P., Tiao, G. C. 1973. Bayesian Inference in Statistical Analysis. Wiley. ISBN 0-471-57428-7.

Breiman, L. 1995. Better subset regression using the nonnegative garrote. Technometrics 37:373–384.

Burgueño, J., G. de los Campos, K. Weigel, and J. Crossa. 2012. Genomic prediction of breeding values when modeling genotype × environment interaction using pedigree and dense molecular markers. Crop Sci. 52(2):707–719. doi:10.2135/cropsci2011.06.0299.

Chawla, Bowyer, Hall, and Kegelmeyer. 2002. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16 (2002) 321–357.

Christoph and José. 2012. On the use of cross-validation for time series predictor evaluation. Information Sciences. doi: 10.1016/j.ins. 2011.12.028.

Delcour, J., and R.C. Hoseney. 2010. Principles of cereal science and technology. 3rd Ed. AACC International, St. Paul, MN. doi:10.1094/9781891127632.

DeLong, R., Elizabeth, David DeLong M., Daniel L. Claeke-Pearson. 1988. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. Biometrics 44, 837-845.

Fawcett, Tom (2006); An introduction to ROC analysis, Pattern Recognition Letters, 27, 861–874.

Friedman J, Hastie T, Tibshirani R. 2010 Regularization paths for generalized linear models via coordinate descent. Journal of Statistical software. 33: 1-22.

Garg, M., H. Singh, H. Kaur, and H.S. Dhaliwal. 2006. Genetic control of high protein content and its association with bread-making quality in wheat. J. Plant Nutr. 29:1357–1369 doi:10.1080/01904160600830134.

González-Recio, O., Rosa, G., J. M., Gianola, D. (2014). Machine learning methods and predictive ability metrics for genome-wide prediction of complex traits. Livest. Sci. 166, 217–231. doi: 10.1016/j.livsci.2014.05.036

Habier, D., Fernando, R.L., and Dekkers, J.C.M.. 2007. The impact of genetic relationship information on genome-assisted breeding values. Genetics 177:2389–2397.

Hand, David J.; and Till, Robert J. 2001; A simple generalization of the area under the ROC curve for multiple class classification problems, Machine Learning, 45, 171–186.

Hayes, B.J., Visscher, P.M., and Goddard, M.E.. 2009. Increased accuracy of artificial selection by using the realized relationship matrix. Genet. Res. Camb. 91:47-60.10.1017/S0016672308009981.

Heffner, E.L., M.E. Sorrells, and J.L. Jannink. 2009. Genomic selection for crop improvement. Crop Sci. 49:1 doi:10.2135/cropsci2008.08.0512.

Heffner, E.L., J.L. Jannink, and M.E. Sorrells. 2011. Genomic selection accuracy using multifamily prediction models in a wheat breeding program. Plant Genome 4:65–75 doi:10.3835/plantgenome.2010.12.0029.

Henderson, C.R. 1975. "Best linear unbiased estimation and prediction under a selection model". Biometrics. 31 (2): 423–447. doi:10.2307/2529430. JSTOR 2529430. PMID 1174616.

Heslot, N., D. Akdemir, M.E. Sorrells, and J.-L. Jannink. 2014. Integrating environmental covariates and crop modeling into the genomic selection framework to predict genotype by environment interactions. Teor. Appl. Genet. 127(2):463–480. doi:10.1007/s00122-013-2231-5.

Ho, Tin Kam 1995. Random Decision Forests. Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282.

Isidro, J., J.-L. Jannink, D. Akdemir, J. Poland, N. Heslot, and M.E. Sorrells. 2015. Training set optimization under population structure in genomic selection. Teor. Appl. Genet. 128:145–158. doi:10.1007/s00122-014-2418-4.

Jarquín, D., J. Crossa, X. Lacaze, P. Du Cheyron, J. Daucourt, J. Lorgeou, et al. 2014. A reaction norm model for genomic selection using highdimensional genomic and environmental data. Teor. Appl. Genet. 127(3):595–607. doi:10.1007/s00122-013-2243-1.

Jeffrey B. Endelman. 2011 Ridge regression and other kernels for genomic selection with R Package rrBLUP. The Plant Genome. https://doi.org/10.3835/plantgenome2011.08.0024.

Kang, H.M., Zaitlen, N.A., Wade, C.M., Kirby, A., Heckerman, D., Daly, M.J., and Eskin, E.. 2008. Efficient control of population structure in model organism association mapping. Genetics 178:1709–1723.10.1534/genetics.107.080101.

Lado, B., P.G. Barrios, M. Quincke, P. Silva, and L. Gutiérrez. 2016. Modeling genotype × environment interaction for genomic selection with unbalanced data from a wheat breeding program. Crop Sci. 56:1–15. doi:10.2135/cropsci2015.04.0207.

Liu, H., H. Zhou, Y. Wu, X. Li, J. Zhao, T. Zuo, et al. 2015. Te impact of genetic relationship and linkage disequilibrium on genomic selection. PLoS ONE 10(7):1–13.

Lorenzana RE, Bernardo R 2009. Accuracy of genotypic value predictions for marker-based selection in biparental plant populations. Theor Appl Genet 120:151–161.

Ma, W., Sutherland, M.W., Kammholz, S., Banks, P., Brennan, P., Bovill, W., & Daggard, G. 2007. Wheat Flour Protein Content and Water Absorption Analysis in a Doubled Haploid Population. Journal of Cereal Science, 45 (3), 302–308.

Ma, Z., Dai, Q. 2016. Selected an Stacking ELMs for Time Series Prediction. Neural Process Lett 44, 831–856 2016. https://doi.org/10.1007/s11063-016-9499-9

Meuwissen, T.H., B.J. Hayes, and M.E. Goddard. 2001. Prediction of total genetic value using genome-wide dense marker maps. Genetics 157(4):1819–1829.

Payne, P.I., M.A. Nightingale, A.F. Krattiger, and L.M. Holt. 1987. Te relationship between HMW glutenin subunit composition and the bread-making quality of British-grown wheat varieties. J. Sci. Food Agric.40:51–65. doi:10.1002/jsfa.2740400108.

Piepho H-P: Ridge regression and extensions for genomewide selection in maize. Crop Science 2009, 49:1165-1176.

Poland, J.A., and T.W. Rife. 2012. Genotyping-by-sequencing for plant breeding and genetics. Plant Genome 5:92 doi:10.3835/plantgenome2012.05.0005.

Rahman, R., Dhruba, S.R., Ghosh, S. et al. Functional random forest with applications in dose-response predictions. Sci Rep 9, 1628 (2019). DOI: https://doi.org/10.1038/s41598-018-38231-w.

Singh, R. Paul and Kent-Jones, Douglas W. 2021. Cereal processing. Encyclopedia Britannica. https://www.britannica.com/technology/cereal-processing.

Whittaker, J.C., R. Thompson, and M.C. Denham. 2000. Marker-assisted selection using ridge regression. Genet. Res. 75:249–252.

Xu, S. 2003. Estimating polygenic effects using markers of the entire genome. Genetics 163:789–801.

Zhao, Y., J. Zeng, R. Fernando, and J.C. Reif. 2013. Genomic prediction of hybrid wheat performance. Crop Sci. 53(3):802–810. doi:10.2135/cropsci2012.08.0463.

Zheng, S., P.F. Byrne, G. Bai, X. Shan, S.D. Reid, S.D. Haley, et al. 2009. Association analysis reveals effects of wheat glutenin alleles and rye translocations on dough-mixing properties. J. Cereal Sci. 50:283–290. doi: 10.1016/j.jcs.2009.06.008.

APPENDIX A. R CODES FOR LINEAR MODEL

```
library(readxl)
data <- read_excel("data.xlsx")
par(mfrow=c(1,1))
plot(data$TW,data$volume)
plot(data$Protein,data$volume)
cor(scale(data$Protein),data$volume,use="complete.obs")
hist(data$Protein)
hist(data$volume)
plot(data$year,data$volume)
#check outliers
boxplot(data$volume,main="volume")
boxplot(data$TW,main="TW")
boxplot(data$Protein,main="Protein")
boxplot(data$volume,main="Volume")
boxplot(data$Extraction,main="Extraction")
boxplot(data$Mixograph,main="Mixograph")
##data11
train=data[data$year==2011,]
test=data[data$year==2012,]
#test=test[,c(1:7)]
fittedModel1=glm(volume~TW+Protein,family=gaussian,data=train)
#summary(fittedModel1)
```

```
AIC(fittedModel1)

fittedvolume=predict(fittedModel1,newdata=test)

sse=sum((test$volume-fittedvolume)^2)

sst=sum((test$volume-mean(test$volume))^2)

1-sse/sst

cor(fittedvolume,test$volume)

cor(rank(fittedvolume),rank(test$volume))

data12=cbind(test,fittedvolume)

##data12

train=data[data$year==2011 | data$year==2012,]

test=data[data$year==2013,]

fittedModel1=glm(volume~TW*Protein+year,family=gaussian,data=train)

#summary(fittedModel1)

fittedvolume=predict(fittedModel1,newdata=test)

cor(fittedvolume,test$volume)

data13=cbind(test,fittedvolume)

##data14

train=data[data$year==2011 | data$year==2012 | data$year==2013,]

test=data[data$year==2014,]

fittedModel1=glm(volume~TW*Protein+year,family=gaussian,data=train)

#summary(fittedModel1)

fittedvolume=predict(fittedModel1,newdata=test)

cor(fittedvolume,test$volume)
```

```
data14=cbind(test,fittedvolume)
##data15
train=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014,]
test=data[data$year==2015,]
fittedModel1=glm(volume~TW*Protein+year,family=gaussian,data=train)
#summary(fittedModel1)
#predict(test$volume,fittedModel1)
fittedvolume=predict(fittedModel1,newdata=test)
cor(fittedvolume,test$volume)
data15=cbind(test,fittedvolume)
##data16
train=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014 |
data$year==2015,]
test=data[data$year==2016,]
#train=train[complete.cases(train),]
test=test[complete.cases(test),]
fittedModel1=glm(volume~TW*Protein+year,family=gaussian,data=train)
#summary(fittedModel1)
#predict(test$volume,fittedModel1)
fittedvolume=predict(fittedModel1,newdata=test)
cor(fittedvolume,test$volume)
data16=cbind(test,fittedvolume)
##data18
```

```
    train=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014 |
data$year==2015 | data$year==2016,]
    test=data[data$year==2018,]
    #train=train[complete.cases(train),]
    test=test[complete.cases(test),]
    fittedModel1=glm(volume~TW*Protein+year,family=gaussian,data=train)
    #summary(fittedModel1)
    #predict(test$volume,fittedModel1)
    fittedvolume=predict(fittedModel1,newdata=test)
    cor(fittedvolume,test$volume)
    data18=cbind(test,fittedvolume)
    data_lm_volume=rbind(data12,data13,data14,data15,data16,data18)
    library(xlsx)
    write.xlsx(data_lm_volume,file="data_lm_volume.xlsx",sheetName = "data_lm_volume",
append = FALSE)
```

APPENDIX B. R CODES FOR BAYESIAN LINEAR MODEL

```
library(readxl)
data <- read_excel("data.xlsx")
library(BAS)
bay=function(train,test){
  fit= bas.lm(volume ~TW+Protein, data = train, prior = "BIC",
              modelprior = Bernoulli(1),
              include.always = ~ .,
              n.models = 1)
  fittedvolume=predict(fit, newdata=test, estimator="BPM", se.fit=TRUE)$fit
  return(fittedvolume)
}
###12
train=data[data$year==2011,]
test=data[data$year==2012,]
fittedvolume=bay(train,test)
cor(fittedvolume,test$volume)
cor(rank(fittedvolume),rank(test$volume))
data12=cbind(test,fittedvolume)
##data13
train=data[data$year==2011 | data$year==2012,]
test=data[data$year==2013,]
fittedvolume=bay(train,test)
```

```
cor(fittedvolume,test$volume)
cor(rank(fittedvolume),rank(test$volume))
data13=cbind(test,fittedvolume)
##data14
train=data[data$year==2011 | data$year==2012 | data$year==2013,]
test=data[data$year==2014,]
fittedvolume=bay(train,test)
cor(fittedvolume,test$volume)
cor(rank(fittedvolume),rank(test$volume))
data14=cbind(test,fittedvolume)
##data15
train=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014,]
test=data[data$year==2015,]
fittedvolume=bay(train,test)
cor(fittedvolume,test$volume)
cor(rank(fittedvolume),rank(test$volume))
data15=cbind(test,fittedvolume)
##data16
train=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014 |
data$year==2015,]
test=data[data$year==2016,]
fittedvolume=bay(train,test)
cor(fittedvolume,test$volume)
```

```
    cor(rank(fittedvolume),rank(test$volume))
    data16=cbind(test,fittedvolume)
    ##data18
    train=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014 |
data$year==2015 | data$year==2016,]
    test=data[data$year==2018,]
    train=train[complete.cases(train),]
    fittedvolume=bay(train,test)
    cor(fittedvolume,test$volume)
    cor(rank(fittedvolume),rank(test$volume))
    data18=cbind(test,fittedvolume)
    data_bay_volume=rbind(data12,data13,data14,data15,data16,data18)
    library(xlsx)
write.xlsx(data_bay_volume,file="data_bay_volume.xlsx",sheetName = "data_bay_volume",
append = FALSE)
```

APPENDIX C. R CODES FOR K NEAREST NEIGHBOR

```
library(class)
library(readxl)
library(caret)
library(ROSE)
data <- read_excel("data.xlsx")
##data12
train12=data[data$year==2011,]
test12=data[data$year==2012,]
##data13
train13=data[data$year==2011 | data$year==2012,]
test13=data[data$year==2013,]
##data14
train14=data[data$year==2011 | data$year==2012 | data$year==2013,]
test14=data[data$year==2014,]
##data15
train15=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014,]
test15=data[data$year==2015,]
##data16
train16=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014 |
data$year==2015,]
test16=data[data$year==2016,]
##data18
```

```
train18=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014 |
data$year==2015 | data$year==2016,]
test18=data[data$year==2018,]
######
findk=function(Train,Test,train_labels){
 i=1
 k.optm=1
 for (i in 1:10){
  knn.mod <- knn(train=Train, test=Test, cl=train_labels, k=i)
  k.optm[i] <- 100 * sum(test_labels == knn.mod)/NROW(test_labels)
  k=i
  cat(k,'=',k.optm[i],'
      ')
 }
 plot(k.optm, type="b", xlab="K- Value",ylab="Accuracy level")
 }
######
data.rose<- ROSE(Group_V~TW+Protein, data = train12, seed = 3)$data
Test12=cbind(test12$Protein,test12$TW)
Train12=cbind(data.rose$Protein,data.rose$TW)
train_labels=data.rose$Group_V
test_labels=test12$Group_V
findk(Train12,Test12,train_labels)
```

```
knn<- knn(train=Train12, test=Test12, cl=train_labels, k=4)
ACC<- 100 * sum(test_labels == knn)/NROW(test_labels)
table(knn,test_labels)
test12$Group_V_knn=knn
######
data.rose<- ROSE(Group_V~TW+Protein, data = train13, seed = 3)$data
Test13=cbind(test13$Protein,test13$TW)
Train13=cbind(data.rose$Protein,data.rose$TW)
train_labels=data.rose$Group_V
test_labels=test13$Group_V
findk(Train13,Test13,train_labels)
knn<- knn(train=Train13, test=Test13, cl=train_labels, k=7)
ACC<- 100 * sum(test_labels == knn)/NROW(test_labels)
table(knn,test_labels)
test13$Group_V_knn=knn
########
data.rose<- ROSE(Group_V~TW+Protein, data = train14, seed = 3)$data
Test14=cbind(test14$Protein,test14$TW)
Train14=cbind(data.rose$Protein,data.rose$TW)
train_labels=data.rose$Group_V
test_labels=test14$Group_V
findk(Train14,Test14,train_labels)
```

```
knn<- knn(train=Train14, test=Test14, cl=train_labels, k=4)
ACC<- 100 * sum(test_labels == knn)/NROW(test_labels)
table(knn,test_labels)
test14$Group_V_knn=knn
#########
data.rose<- ROSE(Group_V~TW+Protein, data = train15, seed = 3)$data
Test15=cbind(test15$Protein,test15$TW)
Train15=cbind(data.rose$Protein,data.rose$TW)
train_labels=data.rose$Group_V
test_labels=test15$Group_V
findk(Train15,Test15,train_labels)
knn<- knn(train=Train15, test=Test15, cl=train_labels, k=4)
ACC<- 100 * sum(test_labels == knn)/NROW(test_labels)
table(knn,test_labels)
test15$Group_V_knn=knn
#####
data.rose<- ROSE(Group_V~TW+Protein, data = train16, seed = 3)$data
Test16=cbind(test16$Protein,test16$TW)
Train16=cbind(data.rose$Protein,data.rose$TW)
train_labels=data.rose$Group_V
test_labels=test16$Group_V
findk(Train16,Test16,train_labels)
```

```
knn<- knn(train=Train16, test=Test16, cl=train_labels, k=10)
ACC<- 100 * sum(test_labels == knn)/NROW(test_labels)
table(knn,test_labels)
test16$Group_V_knn=knn
######
data.rose<- ROSE(Group_V~TW+Protein, data = train18, seed = 3)$data
test18=test18[complete.cases(test18$volume),]
Test18=cbind(test18$Protein,test18$TW)
Train18=cbind(data.rose$Protein,data.rose$TW)
train_labels=data.rose$Group_V
test_labels=test18$Group_V
findk(Train18,Test18,train_labels)
knn<- knn(train=Train18, test=Test18, cl=train_labels, k=2)
ACC<- 100 * sum(test_labels == knn)/NROW(test_labels)
table(knn,test_labels)
test18$Group_V_knn=knn
###
data_knn_volume=rbind(test12,test13,test14,test15,test16,test18)
library(xlsx)
write.xlsx(data_knn_volume,file="data_knn_volume.xlsx",sheetName = "data_knn_volume",
append = FALSE)
```

APPENDIX D. R CODES FOR RANDOM FOREST

```
library(readxl)
library(ROSE)
library(randomForest)
data <- read_excel("data.xlsx")
##data12
train12=data[data$year==2011,]
test12=data[data$year==2012,]
##data13
train13=data[data$year==2011 | data$year==2012,]
test13=data[data$year==2013,]
##data14
train14=data[data$year==2011 | data$year==2012 | data$year==2013,]
test14=data[data$year==2014,]
##data15
train15=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014,]
test15=data[data$year==2015,]
##data16
train16=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014 |
data$year==2015,]
test16=data[data$year==2016,]
##data18
```

```
    train18=data[data$year==2011 | data$year==2012 | data$year==2013 | data$year==2014 |
data$year==2015 | data$year==2016,]
    test18=data[data$year==2018,]
    ######
    data.rose<- ROSE(Group_V~TW+Protein, data = train12, seed = 3)$data
    Test12=test12[,c(4,5)]
    Train12=data.rose
    Train12$Group_V=factor(Train12$Group_V)
    test_labels=test12$Group_V
    rf <- randomForest(Group_V ~ .,data=Train12)
    pred = predict(rf, newdata=Test12)
    ACC<- 100 * sum(test_labels == pred)/NROW(test_labels)
    table(pred,test_labels)
    test12$Group_V_rf=pred
    ######
    data.rose<- ROSE(Group_V~TW+Protein, data = train13, seed = 3)$data
    Test13=test13[,c(4,5)]
    Train13=data.rose
    Train13$Group_V=factor(Train13$Group_V)
    test_labels=test13$Group_V
    rf <- randomForest(Group_V ~ .,data=Train13)
    pred = predict(rf, newdata=Test13)
    ACC<- 100 * sum(test_labels == pred)/NROW(test_labels)
```

```
table(pred,test_labels)

test13$Group_V_rf=pred

########

data.rose<- ROSE(Group_V~TW+Protein, data = train14, seed = 3)$data

Test14=test14[,c(4,5)]

Train14=data.rose

Train14$Group_V=factor(Train14$Group_V)

test_labels=test14$Group_V

rf <- randomForest(Group_V ~ .,data=Train14)

pred = predict(rf, newdata=Test14)

ACC<- 100 * sum(test_labels == pred)/NROW(test_labels)

table(pred,test_labels)

test14$Group_V_rf=pred

#########

data.rose<- ROSE(Group_V~TW+Protein, data = train15, seed = 3)$data

Test15=test15[,c(4,5)]

Train15=data.rose

Train15$Group_V=factor(Train15$Group_V)

test_labels=test15$Group_V

rf <- randomForest(Group_V ~ .,data=Train15)

pred = predict(rf, newdata=Test15)

ACC<- 100 * sum(test_labels == pred)/NROW(test_labels)

table(pred,test_labels)
```

```
test15$Group_V_rf=pred
#####
data.rose<- ROSE(Group_V~TW+Protein, data = train16, seed = 3)$data
Test16=test16[,c(4,5)]
Train16=data.rose
Train16$Group_V=factor(Train16$Group_V)
test_labels=test16$Group_V
rf <- randomForest(Group_V ~ .,data=Train16)
pred = predict(rf, newdata=Test16)
ACC<- 100 * sum(test_labels == pred)/NROW(test_labels)
table(pred,test_labels)
test16$Group_V_rf=pred
######
data.rose<- ROSE(Group_V~TW+Protein, data = train18, seed = 3)$data
Test18=test18[,c(4,5)]
Train18=data.rose
Train18$Group_V=factor(Train18$Group_V)
test_labels=test18$Group_V
rf <- randomForest(Group_V ~ .,data=Train18)
pred = predict(rf, newdata=Test18)
ACC<- 100 * sum(test_labels == pred)/NROW(test_labels)
table(pred,test_labels)
test18$Group_V_rf=pred
```

```
    ###
    data_rf_volumn=rbind(test12,test13,test14,test15,test16,test18)
    library(xlsx)
    write.xlsx(data_rf_volumn,file="data_rf_volumn.xlsx",sheetName = "data_rf_volumn",
append = FALSE)
```

APPENDIX E. R CODES FOR G-BLUP

```
library(xlsx)

library(readxl)

BLUP.AYT11_16<- read_excel("blups_18.xlsx")

BLUP.AYT11_16[1:3,]

library(MASS)

library(psych)

BLUP.AYT11=BLUP.AYT11_16[1:112,]

dim(BLUP.AYT11)

BLUP.AYT11[1:3,]

BLUP.AYT12=BLUP.AYT11_16[113:184,]

dim(BLUP.AYT12)

BLUP.AYT12[1:3,]

BLUP.AYT13=BLUP.AYT11_16[185:215,]

dim(BLUP.AYT13)

BLUP.AYT13[1:3,]

BLUP.AYT14=BLUP.AYT11_16[216:281,]

dim(BLUP.AYT14)

BLUP.AYT14[1:3,]

BLUP.AYT15=BLUP.AYT11_16[282:346,]

dim(BLUP.AYT15)

BLUP.AYT15[1:3,]

BLUP.AYT16=BLUP.AYT11_16[347:427,]
```

```
dim(BLUP.AYT16)

BLUP.AYT16[1:3,]

BLUP.AYT18=BLUP.AYT11_16[BLUP.AYT11_16$year==2018,]

dim(BLUP.AYT18)

BLUP.AYT18[1:3,]

##AYT12, 13, 14, 15, 16 as training population to predict AYT11

phenotype.train=BLUP.AYT11_16[c(113:427),]

dim(phenotype.train)

phenotype.train[1:3,]

phenotype.valid=BLUP.AYT11_16[1:112,]

dim(phenotype.valid)

phenotype.valid[1:3,]

##AYT11, 13, 14, 15, 16 as training population to predict AYT12

phenotype.train=BLUP.AYT11_16[c(1:112, 185:427),]

dim(phenotype.train)

phenotype.train[1:3,]

phenotype.valid=BLUP.AYT11_16[113:184,]

dim(phenotype.valid)

phenotype.valid[1:3,]

##AYT11, 12, 14, 15, 16 as training population to predict AYT13

phenotype.train=BLUP.AYT11_16[c(1:184, 216:427),]

dim(phenotype.train)

phenotype.train[1:3,]
```

```
phenotype.valid=BLUP.AYT11_16[185:215,]
dim(phenotype.valid)
phenotype.valid[1:3,]
##AYT11, 12, 13, 15, 16 as training population to predict AYT14
phenotype.train=BLUP.AYT11_16[c(1:215, 282:427),]
dim(phenotype.train)
phenotype.train[1:3,]
phenotype.valid=BLUP.AYT11_16[216:281,]
dim(phenotype.valid)
phenotype.valid[1:3,]
##AYT11, 12, 13, 14, 16 as training population to predict AYT15
phenotype.train=BLUP.AYT11_16[c(1:281, 347:427),]
dim(phenotype.train)
phenotype.train[1:3,]
phenotype.valid=BLUP.AYT11_16[282:346,]
dim(phenotype.valid)
phenotype.valid[1:3,]
##AYT11-12-13-14-15 as training population to predict AYT16
phenotype.train=BLUP.AYT11_16[1:346,]
dim(phenotype.train)
phenotype.train[1:3,]
phenotype.valid=BLUP.AYT11_16[347:427,]
dim(phenotype.valid)
```

```
phenotype.valid[1:3,]
##AYT11-12-13-14-15 as training population to predict AYT18
phenotype.train=BLUP.AYT11_16[BLUP.AYT11_16$year!=2018,]
dim(phenotype.train)
phenotype.train[1:3,]
phenotype.valid=BLUP.AYT11_16[BLUP.AYT11_16$year==2018,]
dim(phenotype.valid)
phenotype.valid[1:3,]
GBS50.AYT11_16=read.delim("genotype_AYT11-
16_427GBSna50ABD_LDKNNimp.F01_meanImp.txt", header=T, sep=",", na.string="NA")
dim(GBS50.AYT11_16)
GBS50.AYT11_16[1:5,1:20]
library(readr)
SW_GBS20_21_48_Homo_NA50_F01_imputed <-
read_csv("SW_GBS20_21_48_Homo_NA50_F01_imputed.csv")
GBS50.AYT11=as.matrix(GBS50.AYT11_16[1:112,])
GBS50.AYT12=as.matrix(GBS50.AYT11_16[113:184,])
GBS50.AYT13=as.matrix(GBS50.AYT11_16[185:215,])
GBS50.AYT14=as.matrix(GBS50.AYT11_16[216:281,])
GBS50.AYT15=as.matrix(GBS50.AYT11_16[282:346,])
GBS50.AYT16=as.matrix(GBS50.AYT11_16[347:427,])
GBS50.AYT18=as.matrix(SW_GBS20_21_48_Homo_NA50_F01_imputed[428:477,-1])
```

```
## define train and valid genotype data
Markers_impute.train=as.matrix(rbind(GBS50.AYT12, GBS50.AYT13, GBS50.AYT14,
GBS50.AYT15, GBS50.AYT16))
Markers_impute.valid=as.matrix(GBS50.AYT11)
Markers_impute.train=as.matrix(rbind(GBS50.AYT11, GBS50.AYT13, GBS50.AYT14,
GBS50.AYT15, GBS50.AYT16))
Markers_impute.valid=as.matrix(GBS50.AYT12)
Markers_impute.train=as.matrix(rbind(GBS50.AYT11, GBS50.AYT12, GBS50.AYT14,
GBS50.AYT15, GBS50.AYT16))
Markers_impute.valid=as.matrix(GBS50.AYT13)
Markers_impute.train=as.matrix(rbind(GBS50.AYT11,GBS50.AYT12, GBS50.AYT13,
GBS50.AYT15, GBS50.AYT16))
Markers_impute.valid=as.matrix(GBS50.AYT14)
Markers_impute.train=as.matrix(rbind(GBS50.AYT11, GBS50.AYT12, GBS50.AYT13,
GBS50.AYT14, GBS50.AYT16))
Markers_impute.valid=as.matrix(GBS50.AYT15)
Markers_impute.train=as.matrix(rbind(GBS50.AYT11, GBS50.AYT12, GBS50.AYT13,
GBS50.AYT14, GBS50.AYT15))
Markers_impute.valid=as.matrix(GBS50.AYT16)
Markers_impute.train=as.matrix(SW_GBS20_21_48_Homo_NA50_F01_imputed[1:427,-1])
Markers_impute.valid=as.matrix(GBS50.AYT18)
dim(Markers_impute.train)
dim(Markers_impute.valid)
```

```
    ###############################################################
    library(rrBLUP)
    m_train=Markers_impute.train
    m_pred=Markers_impute.valid
    for (i in c(4,5,7,10)){
     y_train=t(as.matrix(phenotype.train[,i]))
     ans.RR=G.BLUP(y=y_train,G.train=m_train,G.pred=m_pred)
    ans.GAUSS=G.BLUP(y=y_train,G.train=m_train,G.pred=m_pred,K.method='GAUSS')
     ans.EXP=G.BLUP(y=y_train,G.train=m_train,G.pred=m_pred,K.method='EXP')
     #cor(ans.RR$g.pred,phenotype.valid[,i])
     #cor(ans.GAUSS$g.pred,phenotype.valid[,i])
     #cor(ans.EXP$g.pred,phenotype.valid[,i])
     if(i==4){
       ANS_Extraction=phenotype.valid[,c(1,4)]
       ANS_Extraction[,3]=ans.RR$g.pred
       ANS_Extraction[,4]=ans.GAUSS$g.pred
       ANS_Extraction[,5]=ans.EXP$g.pred
names(ANS_Extraction)=c("Entry","Extraction","Extraction_RR","Extraction_G","Extraction_E
XP")
     }
     else if(i==5){
       ANS_Absorption=phenotype.valid[,c(1,5)]
       ANS_Absorption[,3]=ans.RR$g.pred
```

```
    ANS_Absorption[,4]=ans.GAUSS$g.pred
    ANS_Absorption[,5]=ans.EXP$g.pred
  names(ANS_Absorption)=c("Entry","Absorption","Absorption_RR","Absorption_G","Absor
ption_EXP")
   }
   if(i==7){
    ANS_Volume=phenotype.valid[,c(1,7)]
    ANS_Volume[,3]=ans.RR$g.pred
    ANS_Volume[,4]=ans.GAUSS$g.pred
    ANS_Volume[,5]=ans.EXP$g.pred
    names(ANS_Volume)=c("Entry","Volume","Volume_RR","Volume_G","Volume_EXP")
   }
   if(i==10){
    ANS_Mixograph=phenotype.valid[,c(1,10)]
    ANS_Mixograph[,3]=ans.RR$g.pred
    ANS_Mixograph[,4]=ans.GAUSS$g.pred
    ANS_Mixograph[,5]=ans.EXP$g.pred
  names(ANS_Mixograph)=c("Entry","Mixograph","Mixograph_RR","Mixograph_G","Mixogr
aph_EXP")
   }
  }
  cor(ANS_Absorption$Absorption,ANS_Absorption$Absorption_RR)
  cor(ANS_Absorption$Absorption,ANS_Absorption$Absorption_G)
```

```
cor(ANS_Absorption$Absorption,ANS_Absorption$Absorption_EXP)
cor(ANS_Extraction$Extraction,ANS_Extraction$Extraction_RR)
cor(ANS_Extraction$Extraction,ANS_Extraction$Extraction_G)
cor(ANS_Extraction$Extraction,ANS_Extraction$Extraction_EXP)
cor(ANS_Mixograph$Mixograph,ANS_Mixograph$Mixograph_RR)
cor(ANS_Mixograph$Mixograph,ANS_Mixograph$Mixograph_G)
cor(ANS_Mixograph$Mixograph,ANS_Mixograph$Mixograph_EXP)
cor(ANS_Volume$Volume,ANS_Volume$Volume_RR)
cor(ANS_Volume$Volume,ANS_Volume$Volume_G)
cor(ANS_Volume$Volume,ANS_Volume$Volume_EXP)
#BLUPS2011=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-
1],ANS_Volume[,-1])
#BLUPS2012=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-
1],ANS_Volume[,-1])
#BLUPS2013=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-
1],ANS_Volume[,-1])
#BLUPS2014=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-
1],ANS_Volume[,-1])
#BLUPS2015=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-
1],ANS_Volume[,-1])
BLUPS2016=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-
1],ANS_Volume[,-1])
```

```
BLUPS2018=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-
1],ANS_Volume[,-1])
BLUPS11_16=rbind(BLUPS2011,BLUPS2012,BLUPS2013,BLUPS2014,BLUPS2015,BLU
PS2016)
write.xlsx(BLUPS11_16, file = "RR-BLUPS11_16.xlsx", append = FALSE)
write.xlsx(BLUPS2018, file = "G-BLUPS18.xlsx", append = FALSE)
########################################################
#GS model_80vs20 cross-validation 11-16AYT
phenotype=as.matrix(BLUP.AYT11_16)
dim(phenotype)
Markers_impute=as.matrix(GBS50.AYT11_16)
dim(Markers_impute)
cycles=3
for(r in 1:cycles){
 train=as.matrix(sample(1:427,342))
 test=setdiff(1:427, train)
 Pheno_train=phenotype[train,]
 m_train=Markers_impute[train,]
 Pheno_valid=phenotype[test,]
 m_pred=Markers_impute[test,]
 for(i in c(4,5,7,10))
 {
  y_train=as.numeric(Pheno_train[,i])
```

```
    ans.RR=G.BLUP(y=y_train,G.train=m_train,G.pred=m_pred)
    ans.GAUSS=G.BLUP(y=y_train,G.train=m_train,G.pred=m_pred,K.method='GAUSS')
    ans.EXP=G.BLUP(y=y_train,G.train=m_train,G.pred=m_pred,K.method='EXP')
    if(i==4){
      ANS_Extraction=as.data.frame(Pheno_valid[,c(1,4)])
      ANS_Extraction[,3]=ans.RR$g.pred
      ANS_Extraction[,4]=ans.GAUSS$g.pred
      ANS_Extraction[,5]=ans.EXP$g.pred
  names(ANS_Extraction)=c("Entry","Extraction","Extraction_RR","Extraction_G","Extractio
n_EXP")
    }
    else if(i==5){
      ANS_Absorption=as.data.frame(Pheno_valid[,c(1,5)])
      ANS_Absorption[,3]=ans.RR$g.pred
      ANS_Absorption[,4]=ans.GAUSS$g.pred
      ANS_Absorption[,5]=ans.EXP$g.pred
  names(ANS_Absorption)=c("Entry","Absorption","Absorption_RR","Absorption_G","Absor
ption_EXP")
    }
    else if(i==7){
      ANS_Volume=as.data.frame(Pheno_valid[,c(1,7)])
      ANS_Volume[,3]=ans.RR$g.pred
      ANS_Volume[,4]=ans.GAUSS$g.pred
```

```
      ANS_Volume[,5]=ans.EXP$g.pred
      names(ANS_Volume)=c("Entry","Volume","Volume_RR","Volume_G","Volume_EXP")
     }
     else if(i==10){
      ANS_Mixograph=as.data.frame(Pheno_valid[,c(1,10)])
      ANS_Mixograph[,3]=ans.RR$g.pred
      ANS_Mixograph[,4]=ans.GAUSS$g.pred
      ANS_Mixograph[,5]=ans.EXP$g.pred
  names(ANS_Mixograph)=c("Entry","Mixograph","Mixograph_RR","Mixograph_G","Mixograph_EXP")
     }
    }
    if(r==1){
     BLUPSCV1=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-1],ANS_Volume[,-1])
    }
    else if(r==2){
     BLUPSCV2=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-1],ANS_Volume[,-1])
    }
    else if(r==3){
     BLUPSCV3=cbind(ANS_Absorption,ANS_Extraction[,-1],ANS_Mixograph[,-1],ANS_Volume[,-1])
```

```
  }
 }
 write.xlsx(BLUPSCV1, file = "BLUPSCV.xlsx", sheetName="BLUPSCV1",append =
FALSE)
 write.xlsx(BLUPSCV2, file = "BLUPSCV.xlsx", sheetName="BLUPSCV2",append =
RR_BLUPS11_16)
 write.xlsx(BLUPSCV3, file = "BLUPSCV.xlsx", sheetName="BLUPSCV3",append =
RR_BLUPS11_16)
```

APPENDIX F. R CODES FOR ROC CURVES

```
library(readxl)
library(zoo)
library(xlsx)
blups_V <- read_excel("G_bayVS_lm.xlsx",sheet="lm")
blups_V <-blups_V [,-1]
#####################18#############################
blups_V18=blups_V[which(blups_V$Year=="2018"),]
blups_V18=blups_V18[complete.cases(blups_V18$Volume),]
blups_V18=blups_V18[complete.cases(blups_V18$lmVolume),]
blups_V18$rank_Volume<-rank(blups_V18$Volume)
n=nrow(blups_V18)
m=as.integer(n*0.3)
blups_V18 <-blups_V18[order(blups_V18$rank_Volume),]
G=split(blups_V18,c(rep(1,each=m),rep(2,each=n-m)))
#####################16#############################
blups_V16=blups_V[which(blups_V$Year=="2016"),]
blups_V16$rank_Volume<-rank(blups_V16$Volume)
summary(blups_V16)
n=nrow(blups_V16)
m=as.integer(n*0.15)
blups_V16 <-blups_V16[order(blups_V16$rank_Volume),]
G=split(blups_V16,c(rep(1,each=m),rep(2,each=n-m)))
#####################15#############################
```

```
blups_V15=blups_V[which(blups_V$Year=="2015"),]
blups_V15=blups_V15[complete.cases(blups_V15$Volume),]
blups_V15=blups_V15[complete.cases(blups_V15$lmVolume),]
blups_V15$rank_Volume<-rank(blups_V15$Volume)
n=nrow(blups_V15)
m=as.integer(n*0.15)
blups_V15 <-blups_V15[order(blups_V15$rank_Volume),]
G=split(blups_V15,c(rep(1,each=m),rep(2,each=n-m)))
########################14############################
blups_V14=blups_V[which(blups_V$Year=="2014"),]
blups_V14=blups_V14[which(blups_V14$EXP=="13"),]
blups_V14$rank_Volume<-rank(blups_V14$Volume)
summary(blups_V14)
n=nrow(blups_V14)
m=as.integer(n*0.15)
blups_V14 <-blups_V14[order(blups_V14$rank_Volume),]
G=split(blups_V14,c(rep(1,each=m),rep(2,each=n-m)))
########################13############################
blups_V13=blups_V[which(blups_V$Year=="2013"),]
blups_V13=blups_V13[which(blups_V13$EXP=="13"),]
blups_V13$rank_Volume<-rank(blups_V13$Volume)
summary(blups_V13)
n=nrow(blups_V13)
```

```
m=as.integer(n*0.15)
blups_V13 <-blups_V13[order(blups_V13$rank_Volume),]
G=split(blups_V13,c(rep(1,each=m),rep(2,each=n-m)))
########################12############################
blups_V12=blups_V[which(blups_V$Year=="2012"),]
blups_V12=blups_V12[which(blups_V12$EXP=="13"),]
blups_V12$rank_Volume<-rank(blups_V12$Volume)
summary(blups_V12)
n=nrow(blups_V12)
m=as.integer(n*0.15)
blups_V12 <-blups_V12[order(blups_V12$rank_Volume),]
G=split(blups_V12,c(rep(1,each=m),rep(2,each=n-m)))
fun(G,G,G,G,n,m,n,m)
fun=function(G,g,H,h,n,m,N,M){
  G2=G$`1` #bottom X
  G1=G$`2`
  G2 <-G2[order(G2$lmVolume,decreasing =T),]
  m1=n-m
  m1*m
  q=rep(0,m+1)
  for(j in 1:m){
    for(i in 1:m1){
      if (G1$lmVolume[i]>G2$lmVolume[j]){ q[j+1]=q[j+1]+1 }
```

```
   else if(G1$lmVolume[i]==G2$lmVolume[j]) {q[j+1]=q[j+1]+0.5 }
   else if(G1$lmVolume[i]<G2$lmVolume[j]) {q[j+1]=q[j+1]+0 }
  }}
q1=c(0:m)/m
q2=q/m1
smoothingSpline2 = smooth.spline(q1,q2, spar=0.4)
plot(q1,q2,xlim = c(0,1),ylim = c(0,1),col="blue",pch=1,lty=2,xlab="FPR",ylab="TPR")
lines(smoothingSpline2,col="blue")
par(new=TRUE)
######################
g2=g$`1`#X n
g1=g$`2`#Y m
g2 <-g2[order(g2$Volume_RR,decreasing =T),]
M1=N-M # m
Q=rep(0,M+1)
for(i in 1:M1){
  for(j in 1:M){
   if (g1$Volume_RR[i]>g2$Volume_RR[j]){ Q[j+1]=Q[j+1]+1 }
   else if(g1$Volume_RR[i]==g2$Volume_RR[j]) {Q[j+1]=Q[j+1]+0.5 }
   else if(g1$Volume_RR[i]<g2$Volume_RR[j]) {Q[j+1]=Q[j+1]+0 }
  }}
Q1=c(0:M)/M
Q2=Q/M1
```

```
smoothingSpline2 = smooth.spline(Q1,Q2, spar=0.4)
plot(Q1,Q2,xlim = c(0,1),ylim = c(0,1),pch=2,col="red",xlab="",ylab="")
lines(smoothingSpline2,col="red")
par(new=TRUE)
####################
H2=H$`1`#X n
H1=H$`2`#Y m
H2 <-H2[order(H2$Volume_EXP,decreasing =T),]
M1=N-M # m
Q=rep(0,M+1)
for(i in 1:M1){
  for(j in 1:M){
    if (H1$Volume_EXP[i]>H2$Volume_EXP[j]){ Q[j+1]=Q[j+1]+1 }
    else if(H1$Volume_EXP[i]==H2$Volume_EXP[j]) {Q[j+1]=Q[j+1]+0.5 }
    else if(H1$Volume_EXP[i]<H2$Volume_EXP[j]) {Q[j+1]=Q[j+1]+0 }
  }}
Q1=c(0:M)/M
Q2=Q/M1
smoothingSpline2 = smooth.spline(Q1,Q2, spar=0.4)
plot(Q1,Q2,xlim = c(0,1),ylim = c(0,1),pch=3,col="yellow",xlab="",ylab="")
lines(smoothingSpline2,col="yellow")
par(new=TRUE)
####################
```

```
h2=h$`1`#X n
h1=h$`2`#Y m
h2 <-h2[order(h2$Volume_G,decreasing =T),]
M1=N-M # m
Q=rep(0,M+1)
for(i in 1:M1){
  for(j in 1:M){
    if (h1$Volume_G[i]>h2$Volume_G[j]){ Q[j+1]=Q[j+1]+1 }
    else if(h1$Volume_G[i]==h2$Volume_G[j]) {Q[j+1]=Q[j+1]+0.5 }
    else if(h1$Volume_G[i]<h2$Volume_G[j]) {Q[j+1]=Q[j+1]+0 }
  }}
Q1=c(0:M)/M
Q2=Q/M1
smoothingSpline2 = smooth.spline(Q1,Q2, spar=0.4)
plot(Q1,Q2,xlim = c(0,1),ylim = c(0,1),pch=4,col="green",xlab="",ylab="")
lines(smoothingSpline2,col="green")
title(main="2018 Volume ROC Curve for LM and G-BLUP",font.main=4)
legend(0.8,0.7,
c("LM","RR","EXP","G"),col=c("blue","red","yellow","green"),pch=1:4,lty=1:4,box.lwd=0.8)
#abline(v=0.2)
id<-c(1:m+1)
(AUC_GLM <- sum(diff(q1[id])*rollmean(q2[id],2)))
(S_GLM=q2[m+1]*q1[m+1])
```

```
  id2=c(1:M+1)
  (AUC_FP <- sum(diff(Q1[id2])*rollmean(Q2[id2],2)))
  (S_PS=Q2[m+1]*Q1[m+1])
  return(c(AUC_GLM,S_GLM,AUC_FP,S_PS))
}
```

APPENDIX G. R CODES FOR COMPARING AUC

```
library(readxl)
G_VS_lm <- read_excel("G_VS_lm.xlsx")
True12=G_VS_lm[G_VS_lm$Year==2012,]
True13=G_VS_lm[G_VS_lm$Year==2013,]
True14=G_VS_lm[G_VS_lm$Year==2014,]
True15=G_VS_lm[G_VS_lm$Year==2015,]
True16=G_VS_lm[G_VS_lm$Year==2016,]
True18=G_VS_lm[G_VS_lm$Year==2018,]
########################11############################
True18=True18[complete.cases(True18$lmVolume),]
n=nrow(True18)
m=as.integer(n*0.15)
True18$rank_Volume<-rank(True18$Volume)
True18 <-True18[order(True18$rank_Volume),]
g=split(True18,c(rep(1,each=m),rep(2,each=n-m)))
######################16############################
n=nrow(True16)
m=as.integer(n*0.15)
True16$rank_Volume<-rank(True16$Volume)
True16 <-True16[order(True16$rank_Volume),]
g=split(True16,c(rep(1,each=m),rep(2,each=n-m)))
########################15############################
```

```
n=nrow(True15)
m=as.integer(n*0.15)
True15$rank_Volume<-rank(True15$Volume)
True15 <-True15[order(True15$rank_Volume),]
g=split(True15,c(rep(1,each=m),rep(2,each=n-m)))
########################14############################
n=nrow(True14)
m=as.integer(n*0.15)
True14$rank_Volume<-rank(True14$Volume)
True14 <-True14[order(True14$rank_Volume),]
g=split(True14,c(rep(1,each=m),rep(2,each=n-m)))
########################13############################
n=nrow(True13)
m=as.integer(n*0.15)
True13$rank_Volume<-rank(True13$Volume)
True13 <-True13[order(True13$rank_Volume),]
g=split(True13,c(rep(1,each=m),rep(2,each=n-m)))
########################12############################
n=nrow(True12)
m=as.integer(n*0.15)
True12$rank_Volume<-rank(True12$Volume)
True12 <-True12[order(True12$rank_Volume),]
g=split(True12,c(rep(1,each=m),rep(2,each=n-m)))
```

```
##############################################################################
fun_RR(g,g,n,m,n,m)
fun_EXP(g,g,n,m,n,m)
fun_G(g,g,n,m,n,m)
fun_RR=function(G,g,n,m,N,M){
 G2=G$`1` #bottom X
 G1=G$`2`
 m1=n-m
 m1*m
 p=0
 for(i in 1:m1){
  for(j in 1:m){
   if (G1$lmVolume[i]>G2$lmVolume[j]){ p=p+1 }
   else if(G1$lmVolume[i]==G2$lmVolume[j]) {p=p+0.5 }
   else if(G1$lmVolume[i]<G2$lmVolume[j]) {p=p+0 }
  }}
 p
 (P=p/(m1*(m)))
 V10=rep(0,m1)
 for(i in 1:m1){
  for(j in 1:m){
   if (G1$lmVolume[i]>G2$lmVolume[j]){ V10[i]=V10[i]+1 }
   else if(G1$lmVolume[i]==G2$lmVolume[j]) {V10[i]=V10[i]+0.5 }
```

```
   else if(G1$lmVolume[i]<G2$lmVolume[j]) {V10[i]=V10[i]+0 }
  }}
V10=V10/m
V01=rep(0,m)
for(j in 1:m){
  for(i in 1:m1){
    if (G1$lmVolume[i]>G2$lmVolume[j]){ V01[j]=V01[j]+1 }
    else if(G1$lmVolume[i]==G2$lmVolume[j]) {V01[j]=V01[j]+0.5 }
    else if(G1$lmVolume[i]<G2$lmVolume[j]) {V01[j]=V01[j]+0 }
  }}
V01=V01/m1
g2=g$`1`#X n
g1=g$`2`#Y m
M1=N-M # m
ps=0
for(i in 1:M1){
  for(j in 1:M){
    if (g1$Volume_RR[i]>g2$Volume_RR[j]){ ps=ps+1 }
    else if(g1$Volume_RR[i]==g2$Volume_RR[j]) {ps=ps+0.5 }
    else if(g1$Volume_RR[i]<g2$Volume_RR[j]) {ps=ps+0 }
  }}
ps
(Ps=ps/(M1*(M)))#probability
```

```
v10=rep(0,M1)
for(i in 1:M1){
  for(j in 1:M){
    if (g1$Volume_RR[i]>g2$Volume_RR[j]){ v10[i]=v10[i]+1 }
    else if(g1$Volume_RR[i]==g2$Volume_RR[j]) {v10[i]=v10[i]+0.5 }
    else if(g1$Volume_RR[i]<g2$Volume_RR[j]) {v10[i]=v10[i]+0 }
  }}
v10=v10/M
v01=rep(0,M)
for(j in 1:M){
  for(i in 1:M1){
    if (g1$Volume_RR[i]>g2$Volume_RR[j]){ v01[j]=v01[j]+1 }
    else if(g1$Volume_RR[i]==g2$Volume_RR[j]) {v01[j]=v01[j]+0.5 }
    else if(g1$Volume_RR[i]<g2$Volume_RR[j]) {v01[j]=v01[j]+0 }
  }}
v01=v01/M1
######################S10###########
s10_11=0
s10_12=0
s10_21=0
s10_22=0
for(i in 1:m1){
  s10_11=s10_11+(V10[i]-P)*(V10[i]-P)
```

```
}
(s10_11=s10_11/(m1-1))
for(i in 1:m1){
  s10_12=s10_12+(V10[i]-P)*(v10[i]-Ps)
}
(s10_12=s10_12/(m1-1))
for(i in 1:m1){
  s10_21=s10_21+(v10[i]-Ps)*(V10[i]-P)
}
(s10_21=s10_21/(m1-1))
for(i in 1:M1){
  s10_22=s10_22+(v10[i]-Ps)*(v10[i]-Ps)
}
(s10_22=s10_22/(M1-1))
S10=matrix(0,2,2)
S10[1,1]=s10_11
S10[1,2]=s10_12
S10[2,1]=s10_21
S10[2,2]=s10_22
######################S01
s01_11=0
s01_12=0
s01_21=0
```

```
s01_22=0
for(i in 1:m){
  s01_11=s01_11+(V01[i]-P)*(V01[i]-P)
}
(s01_11=s01_11/(m1-1))
for(i in 1:m){
  s01_12=s01_12+(V01[i]-P)*(v01[i]-Ps)
}
(s01_12=s01_12/(m1-1))
for(i in 1:m){
  s01_21=s01_21+(v01[i]-Ps)*(V01[i]-P)
}
(s01_21=s01_21/(m1-1))
for(i in 1:M){
  s01_22=s01_22+(v01[i]-Ps)*(v01[i]-Ps)
}
(s01_22=s01_22/(M1-1))
S01=matrix(0,2,2)
S01[1,1]=s01_11
S01[1,2]=s01_12
S01[2,1]=s01_21
S01[2,2]=s01_22
######################lm-rr##
```

```
  S10
  S01
  S=S10/M1+S01/M
  L=t(matrix(c(1,-1)))
  PP=matrix(c(P,Ps))
  Up=L%*%PP+1.96*(L%*%S%*%t(L))^0.5
  Lo=L%*%PP-1.96*(L%*%S%*%t(L))^0.5
  z=L%*%PP/((L%*%S%*%t(L))^0.5)
  (pvalue=2*pnorm(-abs(z)))
  #X=(L%*%PP/((L%*%S%*%t(L))^0.5))^2
  #pchisq(X, df=1, lower.tail=FALSE)
  return(c(Up,Lo,pvalue))
}
fun_G=function(G,g,n,m,N,M){
  G2=G$`1` #bottom X
  G1=G$`2`
  m1=n-m
  m1*m
  p=0
  for(i in 1:m1){
    for(j in 1:m){
      if (G1$lmVolume[i]>G2$lmVolume[j]){ p=p+1 }
      else if(G1$lmVolume[i]==G2$lmVolume[j]) {p=p+0.5 }
```

```
   else if(G1$lmVolume[i]<G2$lmVolume[j]) {p=p+0 }
  }}
p
(P=p/(m1*(m)))
V10=rep(0,m1)
for(i in 1:m1){
  for(j in 1:m){
    if (G1$lmVolume[i]>G2$lmVolume[j]){ V10[i]=V10[i]+1 }
    else if(G1$lmVolume[i]==G2$lmVolume[j]) {V10[i]=V10[i]+0.5 }
    else if(G1$lmVolume[i]<G2$lmVolume[j]) {V10[i]=V10[i]+0 }
  }}
V10=V10/m
V01=rep(0,m)
for(j in 1:m){
  for(i in 1:m1){
    if (G1$lmVolume[i]>G2$lmVolume[j]){ V01[j]=V01[j]+1 }
    else if(G1$lmVolume[i]==G2$lmVolume[j]) {V01[j]=V01[j]+0.5 }
    else if(G1$lmVolume[i]<G2$lmVolume[j]) {V01[j]=V01[j]+0 }
  }}
V01=V01/m1
g2=g$`1`#X n
g1=g$`2`#Y m
M1=N-M # m
```

```
ps=0
for(i in 1:M1){
  for(j in 1:M){
    if (g1$Volume_G[i]>g2$Volume_G[j]){ ps=ps+1 }
    else if(g1$Volume_G[i]==g2$Volume_G[j]) {ps=ps+0.5 }
    else if(g1$Volume_G[i]<g2$Volume_G[j]) {ps=ps+0 }
  }}
ps
(Ps=ps/(M1*(M)))#probability
v10=rep(0,M1)
for(i in 1:M1){
  for(j in 1:M){
    if (g1$Volume_G[i]>g2$Volume_G[j]){ v10[i]=v10[i]+1 }
    else if(g1$Volume_G[i]==g2$Volume_G[j]) {v10[i]=v10[i]+0.5 }
    else if(g1$Volume_G[i]<g2$Volume_G[j]) {v10[i]=v10[i]+0 }
  }}
v10=v10/M
v01=rep(0,M)
for(j in 1:M){
  for(i in 1:M1){
    if (g1$Volume_G[i]>g2$Volume_G[j]){ v01[j]=v01[j]+1 }
    else if(g1$Volume_G[i]==g2$Volume_G[j]) {v01[j]=v01[j]+0.5 }
    else if(g1$Volume_G[i]<g2$Volume_G[j]) {v01[j]=v01[j]+0 }
```

```
 }}
v01=v01/M1
######################S10###########
s10_11=0
s10_12=0
s10_21=0
s10_22=0
for(i in 1:m1){
 s10_11=s10_11+(V10[i]-P)*(V10[i]-P)
}
(s10_11=s10_11/(m1-1))
for(i in 1:m1){
 s10_12=s10_12+(V10[i]-P)*(v10[i]-Ps)
}
(s10_12=s10_12/(m1-1))
for(i in 1:m1){
 s10_21=s10_21+(v10[i]-Ps)*(V10[i]-P)
}
(s10_21=s10_21/(m1-1))
for(i in 1:M1){
 s10_22=s10_22+(v10[i]-Ps)*(v10[i]-Ps)
}
(s10_22=s10_22/(M1-1))
```

```
S10=matrix(0,2,2)
S10[1,1]=s10_11
S10[1,2]=s10_12
S10[2,1]=s10_21
S10[2,2]=s10_22
######################S01
s01_11=0
s01_12=0
s01_21=0
s01_22=0
for(i in 1:m){
  s01_11=s01_11+(V01[i]-P)*(V01[i]-P)
}
(s01_11=s01_11/(m1-1))
for(i in 1:m){
  s01_12=s01_12+(V01[i]-P)*(v01[i]-Ps)
}
(s01_12=s01_12/(m1-1))
for(i in 1:m){
  s01_21=s01_21+(v01[i]-Ps)*(V01[i]-P)
}
(s01_21=s01_21/(m1-1))
for(i in 1:M){
```

```
  s01_22=s01_22+(v01[i]-Ps)*(v01[i]-Ps)
 }
 (s01_22=s01_22/(M1-1))
 S01=matrix(0,2,2)
 S01[1,1]=s01_11
 S01[1,2]=s01_12
 S01[2,1]=s01_21
 S01[2,2]=s01_22
 ######################lm-rr##
 S10
 S01
 S=S10/M1+S01/M
 L=t(matrix(c(1,-1)))
 PP=matrix(c(P,Ps))
 Up=L%*%PP+1.96*(L%*%S%*%t(L))^0.5
 Lo=L%*%PP-1.96*(L%*%S%*%t(L))^0.5
 z=L%*%PP/((L%*%S%*%t(L))^0.5)
 (pvalue=2*pnorm(-abs(z)))
 #X=(L%*%PP/((L%*%S%*%t(L))^0.5))^2
 #pchisq(X, df=1, lower.tail=FALSE)
 return(c(Up,Lo,pvalue))
}
fun_EXP=function(G,g,n,m,N,M){
```

```
G2=G$`1` #bottom X
G1=G$`2`
m1=n-m
m1*m
p=0
for(i in 1:m1){
  for(j in 1:m){
    if (G1$lmVolume[i]>G2$lmVolume[j]){ p=p+1 }
    else if(G1$lmVolume[i]==G2$lmVolume[j]) {p=p+0.5 }
    else if(G1$lmVolume[i]<G2$lmVolume[j]) {p=p+0 }
  }}
p
(P=p/(m1*(m)))
V10=rep(0,m1)
for(i in 1:m1){
  for(j in 1:m){
    if (G1$lmVolume[i]>G2$lmVolume[j]){ V10[i]=V10[i]+1 }
    else if(G1$lmVolume[i]==G2$lmVolume[j]) {V10[i]=V10[i]+0.5 }
    else if(G1$lmVolume[i]<G2$lmVolume[j]) {V10[i]=V10[i]+0 }
  }}
V10=V10/m
V01=rep(0,m)
for(j in 1:m){
```

```
 for(i in 1:m1){
   if (G1$lmVolume[i]>G2$lmVolume[j]){ V01[j]=V01[j]+1 }
   else if(G1$lmVolume[i]==G2$lmVolume[j]) {V01[j]=V01[j]+0.5 }
   else if(G1$lmVolume[i]<G2$lmVolume[j]) {V01[j]=V01[j]+0 }
 }}
V01=V01/m1
g2=g$`1`#X n
g1=g$`2`#Y m
M1=N-M # m
ps=0
for(i in 1:M1){
 for(j in 1:M){
   if (g1$Volume_EXP[i]>g2$Volume_EXP[j]){ ps=ps+1 }
   else if(g1$Volume_EXP[i]==g2$Volume_EXP[j]) {ps=ps+0.5 }
   else if(g1$Volume_EXP[i]<g2$Volume_EXP[j]) {ps=ps+0 }
 }}
ps
(Ps=ps/(M1*(M)))#probability
v10=rep(0,M1)
for(i in 1:M1){
 for(j in 1:M){
   if (g1$Volume_EXP[i]>g2$Volume_EXP[j]){ v10[i]=v10[i]+1 }
   else if(g1$Volume_EXP[i]==g2$Volume_EXP[j]) {v10[i]=v10[i]+0.5 }
```

```
   else if(g1$Volume_EXP[i]<g2$Volume_EXP[j]) {v10[i]=v10[i]+0 }
  }}
v10=v10/M
v01=rep(0,M)
for(j in 1:M){
  for(i in 1:M1){
    if (g1$Volume_EXP[i]>g2$Volume_EXP[j]){ v01[j]=v01[j]+1 }
    else if(g1$Volume_EXP[i]==g2$Volume_EXP[j]) {v01[j]=v01[j]+0.5 }
    else if(g1$Volume_EXP[i]<g2$Volume_EXP[j]) {v01[j]=v01[j]+0 }
  }}
v01=v01/M1
######################S10###########
s10_11=0
s10_12=0
s10_21=0
s10_22=0
for(i in 1:m1){
  s10_11=s10_11+(V10[i]-P)*(V10[i]-P)
}
(s10_11=s10_11/(m1-1))
for(i in 1:m1){
  s10_12=s10_12+(V10[i]-P)*(v10[i]-Ps)
}
```

```
(s10_12=s10_12/(m1-1))
for(i in 1:m1){
  s10_21=s10_21+(v10[i]-Ps)*(V10[i]-P)
}
(s10_21=s10_21/(m1-1))
for(i in 1:M1){
  s10_22=s10_22+(v10[i]-Ps)*(v10[i]-Ps)
}
(s10_22=s10_22/(M1-1))
S10=matrix(0,2,2)
S10[1,1]=s10_11
S10[1,2]=s10_12
S10[2,1]=s10_21
S10[2,2]=s10_22
######################S01
s01_11=0
s01_12=0
s01_21=0
s01_22=0
for(i in 1:m){
  s01_11=s01_11+(V01[i]-P)*(V01[i]-P)
}
(s01_11=s01_11/(m1-1))
```

```
for(i in 1:m){
  s01_12=s01_12+(V01[i]-P)*(v01[i]-Ps)
}
(s01_12=s01_12/(m1-1))
for(i in 1:m){
  s01_21=s01_21+(v01[i]-Ps)*(V01[i]-P)
}
(s01_21=s01_21/(m1-1))
for(i in 1:M){
  s01_22=s01_22+(v01[i]-Ps)*(v01[i]-Ps)
}
(s01_22=s01_22/(M1-1))
S01=matrix(0,2,2)
S01[1,1]=s01_11
S01[1,2]=s01_12
S01[2,1]=s01_21
S01[2,2]=s01_22
######################lm-rr##
S10
S01
S=S10/M1+S01/M
L=t(matrix(c(1,-1)))
PP=matrix(c(P,Ps))
```

```
  Up=L%*%PP+1.96*(L%*%S%*%t(L))^0.5
  Lo=L%*%PP-1.96*(L%*%S%*%t(L))^0.5
  z=L%*%PP/((L%*%S%*%t(L))^0.5)
  (pvalue=2*pnorm(-abs(z)))
  #X=(L%*%PP/((L%*%S%*%t(L))^0.5))^2
  #pchisq(X, df=1, lower.tail=FALSE)
  return(c(Up,Lo,pvalue))
}
```