The Statistical Use of Digital Phenotypes in Plant Breeding

Haoyuan Zhang

Delft University of Technology

The Statistical Use of Digital Phenotypes in Plant Breeding

by

Haoyuan Zhang

Supervisors:	Alexis Derumigny from Delft University of Technology Hans Daetwyler from Bayer Crop Science
Faculty:	Faculty of Applied Mathematics, TU Delft
Project Duration:	March, 2024 - January, 2025

Summary

Plant digital phenotypes, which are observable performances of traits collected through photos and sensors by computers, are emerging tools to help breeders evaluate plants. The collection of digital phenotypes has made great progress, but how to apply them to breeding is still unclear. This research used quantitative genetic methods and statistical learning algorithms to build a workflow for the relationship between digital and conventional phenotypes. The workflow was applied to blocky pepper and tomato data.

The bi-trait ss-GBLUP was used to estimate genetic correlations between traits, which quantify the genetic relationship between two traits by measuring the proportion of shared genetic variance. A few trait pairs showed high genetic correlations, while most had moderate or low correlations. The bi-trait ss-GBLUP could include the fruit color as a fixed effect. The likelihood ratio test was used to evaluate the impact of fruit color on model performance. Fruit color was useful if at least one trait was about color, otherwise, it was not beneficial. Genetic and Pearson correlations are two correlations of traits. They differed significantly only if both traits were about color.

Genetic correlations guided predictor selection for statistical learning models including linear regression, LASSO regression, random forest, and XGBoost. Linear and LASSO regressions were underfitted, whereas random forest and XGBoost were overfitted. Simpson's paradox led to misleading results in linear models for the color trait, which was resolved by adding a key predictor. Overfitting wasn't due to insufficient training samples, and tuning more hyperparameters didn't address the issue. Despite being overfitted, the random forest model achieved the best overall performance, with only one of the seven conventional traits being unpredicted.

Keywords: Digital Phenotype; Quantitative Genetics; Statistical Learning

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
BLUE	Best linear unbiased estimator
BLUP	Best linear unbiased prediction
ABLUP	BLUP based on pedigree relationship matrix
GBLUP	BLUP based on genomic relationship matrix
ssGBLUP	Single-step GBLUP
MT ssGBLUP	Multi-Trait Single-step GBLUP
IW	Inverse Wishart
CART	Classification and regression tree

Symbols

Symbol	Definition
$\mathbf{Y}$	Dependent variable
$\mathbf{X}$	Fixed effects & Independent variable
$\mathbf{Z}$	Random effects
β	Fixed effect coefficient
u	Random effect coefficient
ε	Errors in the mixed effects model
μ	Intercept of the mixed effects model
N	Number of observations in the mixed effects model
p	Number of fixed effects in the mixed effects model
q	Number of random effects in the mixed effects model
k	Number of traits in the mixed effects model
$\hat{\beta}$	BLUE of fixed effects
$\hat{u}$	BLUP of random effects
$\mathbf{\Omega_{ST}}$	Variance-covariance matrix of random effect coefficients in single-trait model
$\mathbf{\Omega_{MT}}$	Variance-covariance matrix of random effect coefficients in multi-trait model

Symbol	Definition
$\mathbf{\Omega}_{\mathbf{ST}0}$	Additive genetic relationship matrix
$\mathbf{R}$	Variance-covariance matrix of residuals
σ_u^2	Additive genetic variance
σ_ϵ^2	Residual variance
$\mathbf{A}$	Pedigree relationship matrix
$\mathbf{A}_{i,j}$	Pedigree relationship between individuals i and j
$\mathbf{M}$	Marker genotype matrix
$\mathbf{G}$	Genomic relationship matrix
$\mathbf{G}_{i,j}$	Genomic relationship between individuals i and j
$\mathbf{H}$	Combined relationship matrix
$\mathbf{H}_{i,j}$	Combined relationship between individuals i and j
ω	Hyperparameters from prior distribution
σ_z^2	Variance of random effects in Gaussian priors
m	The degree of freedom of IW distribution
$\mathbf{V}$	The scale matrix of IW distribution
$N_{iterations}$	The number of burn in iteration of Gibbs sampler
$\mathbf{T}_{con}$	Conventional trait set
$\mathbf{T}_{digital}$	Digital trait set
$\mathbf{t}_{con}$	A conventional trait
$\mathbf{t}_{digital}$	A digital trait
$\hat{\rho}^{Pearson}$	Estimated Pearson correlation
$\hat{\rho}^{Gen}$	Estimated genetic correlation
$\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$	The Pearson correlation between real and estimated values by predictive models
$\mathcal{D}$	The training dataset
$\mathcal{D}_{bootstrap}$	The bootstrap sample used to grow a particular CART
N_{trees}	The number of CARTs in an ensemble model
$N_{features}$	The number of features in $\mathcal{D}$
$N_{subfeatures}$	The number of features used in each CART
$N_{minsamples}$	The minimum samples in a leaf
$N_{maxdepth}$	The maximum depth of the CART

1

Introduction

For centuries, breeders have spent much of their time collecting measurements of their plants' features in order to breed better varieties. A phenotype, or phenotypic value, is the observable performance of a trait, used to estimate the unknown genotypic value [1]. The phenotype is the most important tool for describing characteristics and genetic resource management. Different from subjective conventional phenotypes, objective digital phenotypes are automatically collected through sensors and cameras [2, 3]. With the application of ScaleCam, the machine to collect digital phenotypes, Bayer Crop Science has a large number of digital phenotypes. The aim of this research is to help breeders use novel digital phenotypes in their breeding decisions. This chapter will provide an introduction to the research by first discussing the background and research gap of plant digital phenotypes, followed by the workflow to guide the use of digital phenotypes and finally, structural outline of the thesis.

1.1. Background and Research Gap of Plant Digital Phenotypes

Plant phenotyping are categorized as qualitative or quantitative. Qualitative data is used to diagnose traits with significant heritability and remain unaffected by environmental changes. In contrast, quantitative data is for traits arising from gene interactions and are significantly influenced by genotype-environment interactions [2, 4].

Traditional plant phenotyping relies on labor-intensive and time-consuming manual measurements. Manual measurements are subjective and error-prone so data accuracy and reliability cannot be guaranteed. Besides, because of the workforce, cost, and other limitations, breeders can only measure limited traits during key stages of plant growth. Therefore, phenotypic changes cannot be fully tracked throughout the plant life cycle. Digital phenotyping addresses these issues and is a powerful tool for measuring plant traits [2, 5].

Modern plant phenotyping uses digital systems and sensor technologies (e.g., sensitive imaging, spectral imaging, robotics, and advanced calculations) to evaluate complex traits like yield, growth period, disease resistance, and other quantitative parameters. Digital imaging analysis rapidly measures plant traits, which is a key technology in plant digital phenotyping. Additionally, digital phenotyping offers the benefits of a consistent framework, accurate outcomes, and straightforward data storage [1, 2, 4].

The objective of this research is to reduce the gap between plant digital phenotype collection and application. Plant phenotyping has been carried out by farmers and breeders for ages. In the past decade, high-throughput phenotyping platforms have become popular for accurately measuring numerous plant traits in controlled environments, handling thousands of plants per study [4]. Digital phenotyping has made great progress in collection, but its application in breeding remains unclear. Therefore, this research investigated the relationship between digital and conventional plant phenotypes and the use of digital phenotypes in plant breeding. Experts from Bayer Crop Science have proposed a rough workflow (Figure 1.1) to find the relationship between conventional and digital traits by quantitative genetics and statistical learning models. This research filled in details of the workflow and applied it to peppers.

Quantitative genetics, also known as statistical or biometrical genetics, uses statistical analysis to detect genetic models in designed populations. It describes genetic and environmental influences averaged over a population in a specific context. The core of quantitative genetics is variability, where individual differences in traits result from unique genetic and environmental factors over time [6]. In the workflow (Figure 1.1), a quantitative genetical method calculated the genetic correlation between conventional and digital traits, representing the proportion of variance shared by traits due to genetic reasons [7].

1.2. Workflow to Guide the Use of Digital Phenotypes

Step 1: Distribution Check

The workflow begins with a conventional trait $\mathbf{t}_{con}$. First, it evaluates $\mathbf{t}_{con}$'s informativeness with the distribution plot. Breeders focus only on informative traits as they are helpful in breeding decisions. If trait values across varieties show no difference, the trait cannot evaluate plant quality.

Step 2: $\hat{\rho}^{Gen}$ from Multi-Trait Single-Step GBLUP

In the second step (Figure 1.2), genetic correlations $\hat{\rho}^{Gen}$s between the $\mathbf{t}_{con}$ and all digital traits $\mathbf{t}_{digital}$s are calculated using multi-trait single-step GBLUP. Trait pairs ($\mathbf{t}_{con}$ and $\mathbf{t}_{digital}$) are sorted into three buckets based on $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$: high, intermediate, and low genetic correlations. Different methods are applied to each bucket: replacement, new index, and no action.

The conventional trait $\mathbf{t}_{con}$ in the first bucket can be replaced by a $\mathbf{t}_{digital}$ with high genetical correlation

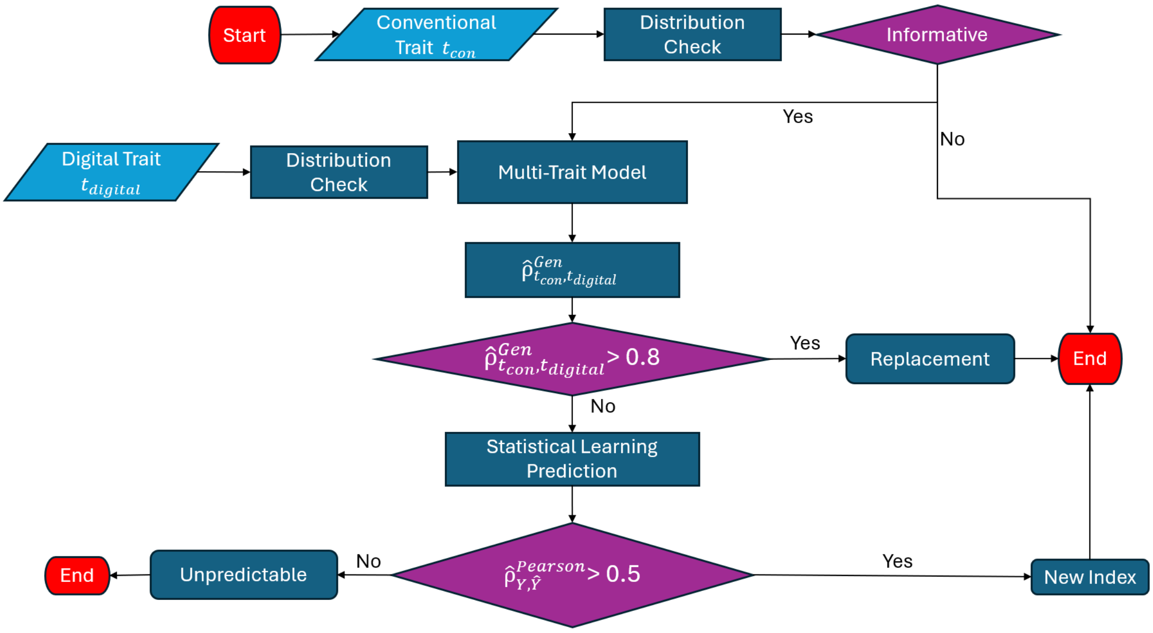
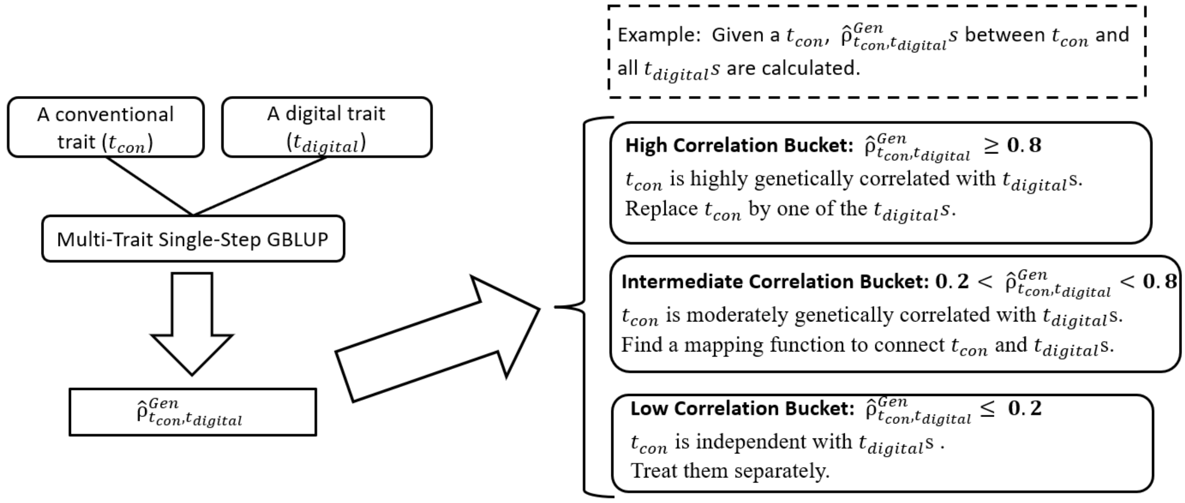

Start
Conventional Trait t_{con}
Distribution Check
Informative
Yes
No
Digital Trait $t_{digital}$
Distribution Check
Multi-Trait Model
$\hat{\rho}^{Gen}_{t_{con},t_{digital}}$
$\hat{\rho}^{Gen}_{t_{con},t_{digital}} > 0.8$
Yes
Replacement
End
No
Statistical Learning Prediction
End
Unpredictable
No
$\hat{\rho}^{Pearson}_{Y,\hat{Y}} > 0.5$
Yes
New Index

Figure 1.1: A workflow for the relationship between digital and conventional traits

with $\mathbf{t}_{con}$. $\mathbf{t}_{con}$s in the second bucket move to Step 3, where statistical models use multiple $\mathbf{t}_{digital}$s to create a new index as its replacement. Trait pairs ($\mathbf{t}_{con}$ and $\mathbf{t}_{digital}$) in the third bucket are independently treated in breeding due to low genetic correlations.


A conventional trait (t_{con})
A digital trait ($t_{digital}$)
Multi-Trait Single-Step GBLUP
$\hat{\rho}^{Gen}_{t_{con},t_{digital}}$

Example: Given a t_{con}, $\hat{\rho}^{Gen}_{t_{con},t_{digital}}$s between t_{con} and all $t_{digital}$s are calculated.

High Correlation Bucket: $\hat{\rho}^{Gen}_{t_{con},t_{digital}} \geq 0.8$
t_{con} is highly genetically correlated with $t_{digital}$s.
Replace t_{con} by one of the $t_{digital}$s.

Intermediate Correlation Bucket: $0.2 < \hat{\rho}^{Gen}_{t_{con},t_{digital}} < 0.8$
t_{con} is moderately genetically correlated with $t_{digital}$s.
Find a mapping function to connect t_{con} and $t_{digital}$s.

Low Correlation Bucket: $\hat{\rho}^{Gen}_{t_{con},t_{digital}} \leq 0.2$
t_{con} is independent with $t_{digital}$s .
Treat them separately.

Figure 1.2: $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ assigns ($\mathbf{t}_{con}$, $\mathbf{t}_{digital}$) pairs to different buckets.
Different methods are applied to each bucket: replacement, new index, and no action.

Step 3: Statistical Learning Prediction

Some $\mathbf{t}_{con}$s cannot be replaced directly by any $\mathbf{t}_{digital}$, but they can be predicted using multiple $\mathbf{t}_{digital}$s. Each $\mathbf{t}_{digital}$ does not have sufficient information to replace $\mathbf{t}_{con}$, but a new index combining several $\mathbf{t}_{digital}$s might succeed. Statistical models (linear regression, LASSO regression, random forest, and XGBoost) map multiple $\mathbf{t}_{digital}$s to one $\mathbf{t}_{con}$ to find a good prediction, the new index. We denote real trait values as $\mathbf{Y}$. The evaluation metric is $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$, Pearson correlation between actual $\mathbf{Y}$ and predicted $\hat{\mathbf{Y}}$. An index of $\mathbf{t}_{digital}$s can replace $\mathbf{t}_{con}$ if $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ exceeds 0.5; otherwise, $\mathbf{t}_{con}$ is unpredictable by $\mathbf{t}_{digital}$s.

Application

Each $\mathbf{t}_{con}$ receives a label from the workflow: noninformative, replacement, new index, or unpredictable. This label guides breeders on collecting $\mathbf{t}_{con}$. Only unpredictable conventional traits need collection, as the digital set can't handle them. It can reduce the workload of breeders to collect phenotypes.

1.3. Structural Outline of the Thesis

This thesis is structured in two main parts: methodology from Chapter 2 to Chapter 5 and application from Chapter 6 to Chapter 9. In the first part, we present theories and models linked to the workflow. Chapter 2 and Chapter 3 first review the linear mixed effects model in general and in breeding, respectively. Chapter 4 describes the Bayesian method used in the linear mixed effects model, followed by the statistical learning models to use in Chapter 5.

In the second part, we present application of the workflow to peppers. Data from peppers is described in Chapter 6. Detailed calculation of $\hat{\rho}^{Gen}$ and prediction of conventional traits are in Chapter 7 and Chapter 8, respectively. Finally we conclude the thesis in Chapter 9.

2

Linear Mixed Effects Model

The simple linear model is a basic regression model expressed as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p. \tag{2.1}$$

Linear mixed effects models extend simple linear models to include fixed and random effects, useful for data with hierarchical structure [8]. For instance, since each plant belongs to one field, they share environmental resources with plants in that field. When studying technology's impact on plant yield, the field should be included as a group-level variable, while yield is an individual-level variable nested within the field. In this chapter, we are going to talk about how to include group-level variables in a regression model using fixed or random effects.

2.1. Fixed and Random Effects

For the fixed effect, we assume a true regression line in the population with slope β, for which we are interested in the estimator $\hat{\beta}$. For random effects, we assume βs are from the same distribution, and we would like to estimate the underlying distribution of β [8].

Another fundamental difference between fixed and random effects is inference/prediction [9]. A fixed effect supports the estimation of only the values of features used in the model. In contrast, a random effect allows us to predict new values about the population from which the sample is drawn. They can be values of the feature that may not have been present now. This is because each level of a random effect can be viewed as a random variable derived from an underlying distribution. By estimating random effects, we can make inferences not just about the specific levels but also about the population level and unobserved levels. This idea, known as exchangeability, suggests that levels within a random effect are not considered separate and independent; instead, they are seen as representative samples from a broader set of levels, some of which may remain unobserved [9].

Best Linear Unbiased Estimates (BLUEs) are the estimates of fixed effects. The linear estimator of β, $\hat{\beta}$, is BLUE if $E(\hat{\beta}) = \beta$ and $Var(\hat{\beta})$ is the minimum among all unbiased estimates [8]. Best Linear Unbiased Predictions (BLUPs) are the predictions of random effects. The linear predictor of u, $\hat{u}$, is BLUP if $E(\hat{u} - u) = 0$ and $Var(\hat{u} - u)$ is the minimum among all unbiased predictions [8]. $\hat{u}$ is a predictor because u is a random variable and there is no true value of u. The definitions of BLUE and BLUP are different because it is assumed β is a constant and u is a random variable. Therefore, $E(\hat{\beta}-\beta) = E(\hat{\beta}) - E(\beta) = E(\hat{\beta}) - \beta$ while $E(\hat{u}-u) = E(\hat{u}) - E(u) \neq E(\hat{u}) - u$. Similarly, $Var(\hat{\beta}-\beta) = Var(\hat{\beta})$ while $Var(\hat{u} - u) \neq Var(\hat{u})$. The calculation of BLUE and BLUP will be covered in Section 2.2.

Consider a hypothetical research study to understand fixed and random effects. The study aims to examine the impact of cultivars on the yield. In plant breeding a cultivar is a kind of plant that people have selected for desired traits [10]. A regression model is used where Y is the yield, X is a categorical variable for the cultivar, and e is the error term. We start with fixed effects. If there are three cultivars, the model can be written as:

$$Y = \beta_0 + \beta_1\mathbf{I}_{\{X=cultivar1\}} + \beta_2\mathbf{I}_{\{X=cultivar2\}} + \beta_3\mathbf{I}_{\{X=cultivar3\}} + e. \tag{2.2}$$

The fixed effect variables capture the individual characteristics of each cultivar that affect yield. Let β_1, β_2 and β_3 be 3, 5 and 7 respectively and the distribution of the error term be $\mathcal{N}(0, 0.1)$. We simulated 100 samples for each cultivar, 300 samples in total, to fit this model. Table 2.1 displays the estimates. It allows for comparing average yields between cultivars but cannot infer about a new cultivar not in the model.

	$\hat{\beta}$	SD($\hat{\beta}$)	t	P-value	95% CI
β_0	3.7422	0.005	822.907	0.000	[3.733,3.751]
β_1	-0.7494	0.009	-86.062	0.000	[-0.767,-0.732]
β_2	1.2395	0.009	142.343	0.000	[1.222,1.257]
β_3	3.2521	0.009	373.468	0.000	[3.235,3.269]

Table 2.1: Output of the fixed model

With 100 cultivars as fixed effects, the model is expressed as

$$Y = \beta_0 + \sum_{j=1}^{100} \beta_j\mathbf{I}_{\{X=cultivar_j\}} + e. \tag{2.3}$$

The output table includes estimates of 100 coefficients and one intercept. In addition to fixed effects, the cultivar can be treated as a random effect if we want to know more about the distribution of β_j. In this case, we assume that all β_js are from a distribution and the regression equation can be written as:

$$Y_{ij} = \beta_0 + \beta_j + e_{ij}, \tag{2.4}$$

where i indexes individual fruits, and $j \in \{1,2,...,100\}$ indexes cultivars. In this case, we estimate the distribution, not the value of β_j. Using $\mathcal{N}(5,4)$, we generated 100 β_js and 100 Y_{ij}s per β_j. Figure 2.1 plots the non-parametrically estimated distribution of β, which has a variance of 4.22. Therefore, 4.22 is typical deviation of new cultivar compared with the average cultivar and we can make prediction of a new cultivar even if it is not used in the model.

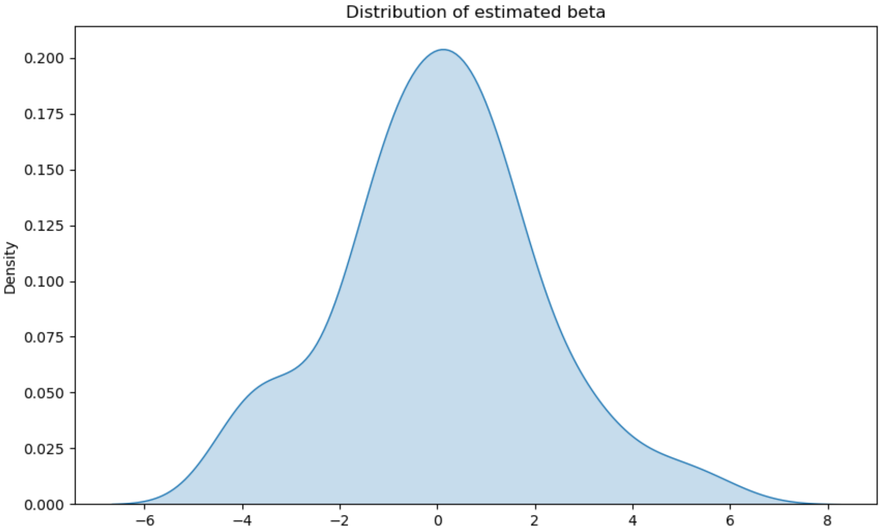


Figure 2.1: Non-parametrically estimated distribution of β

2.2. Linear Mixed Effects Model

A model with only fixed effects is a fixed effects model; with only random effects, it's a random effects model; and with both, it's a mixed effects model. The linear mixed effects model described by Laird and Ware (1982) [11] as:

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{Z}u + \varepsilon \tag{2.5}$$

where

- $\mathbf{Y}$ is a $N \times 1$ vector of dependent variables;
- $\mathbf{X}$ is a $N \times p$ matrix of fixed effects;
- β is a $p \times 1$ vector of fixed effect coefficients;
- $\mathbf{Z}$ is a $N \times q$ matrix of random effects;
- u is a $q \times 1$ vector of random effect coefficients;
- ε is a $N \times 1$ vector of errors. It is the part of $\mathbf{Y}$ that is not explained by $\mathbf{X}\beta + \mathbf{Z}u$.

We now present the linear mixed effects model with the dimensions of variables:

$$\overbrace{\mathbf{Y}}^{\text{N x 1}} = \overbrace{\underbrace{\mathbf{X}}_{\text{N x p}} \quad \underbrace{\beta}_{\text{p x 1}}}^{\text{N x 1}} + \overbrace{\underbrace{\mathbf{Z}}_{\text{N x q}} \quad \underbrace{u}_{\text{q x 1}}}^{\text{N x 1}} + \overbrace{\varepsilon}^{\text{N x 1}}.$$

The random effect coefficients u are defined to have a mean of 0 and therefore any nonzero mean for a term in the random effects is expressed as part of the fixed effect. This is because random effects are modeled as deviations from the fixed effect. It is assumed that $u \sim \mathcal{N}(\mathbf{0}, \mathbf{\Omega_{ST}})$, where $\mathbf{\Omega_{ST}}$ is the variance-covariance matrix of random effect coefficients and "$\boldsymbol{ST}$" is short for "Single Trait" to distinguish it from multi-trait model in Section 3.5. Therefore, it is symmetric and positive semi-definite. Suppose that we had three cultivars, then

$$\mathbf{\Omega_{ST}} = \begin{bmatrix} \sigma^2_{cultivar1} & \text{Cov}(cultivar1, cultivar2) & \text{Cov}(cultivar1, cultivar3) \\ \text{Cov}(cultivar1, cultivar2) & \sigma^2_{cultivar2} & \text{Cov}(cultivar2, cultivar3) \\ \text{Cov}(cultivar1, cultivar3) & \text{Cov}(cultivar2, cultivar3) & \sigma^2_{cultivar3} \end{bmatrix}.$$

Assumption 2.1. *The distribution of random effect coefficients (u) is $\mathcal{N}(\mathbf{0}, \mathbf{\Omega_{ST}})$.*

Various constraints allow us to simplify the model, for example, by assuming that the random effects are independent, which would imply the structure is

$$\mathbf{\Omega_{ST}} = \begin{bmatrix} \sigma^2_{cultivar1} & 0 & 0 \\ 0 & \sigma^2_{cultivar2} & 0 \\ 0 & 0 & \sigma^2_{cultivar3} \end{bmatrix}.$$

Another element in the model is the residual with assumption $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$, where $\mathbf{R}$ is the variance-covariance matrix of residuals. One common structure is $\mathbf{R} = \mathbf{I}\sigma^2_\varepsilon$, where σ^2_ε is the residual variance. This structure assumes a homogeneous residual variance for all observations and residuals are independent [9]. Besides, it is also assumed the algebraic independence between $\mathbf{\Omega_{ST}}$ and $\mathbf{R}$. The joint distribution of u and ε is as follows:

$$\begin{bmatrix} u \\ \varepsilon \end{bmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{\Omega_{ST}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix}\right]. \tag{2.6}$$

Assumption 2.2. *The distribution of error terms (ε) is $\mathcal{N}(\mathbf{0}, \mathbf{R})$.*

Assumption 2.3. *u and ε are independent.*

With Assumptions (2.1, 2.2 and 2.3), the distribution of $\mathbf{Y} \mid \mathbf{X}, \mathbf{Z}$ can be expressed as: $\mathcal{N}(\mathbf{X}\beta, \mathbf{Z}\mathbf{\Omega_{ST}}\mathbf{Z}' + \mathbf{R})$.

According to Henderson (1959) [12], the BLUE $\hat{\beta}$ and BLUP $\hat{u}$ are solutions to the mixed model equations:

$$\begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Y} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y} \end{bmatrix}. \tag{2.7}$$

Proof. Recall that for $\boldsymbol{y} = (y_1, y_2, \dots, y_i)^T \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}_y)$, the probability density function (PDF) is

$$P(\boldsymbol{y}) = (2\pi)^{-\frac{i}{2}} \left|\boldsymbol{\Sigma}_y\right|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}_y^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right]. \tag{2.8}$$

Combine Equation (2.6) and (2.8). We have

$$\begin{aligned} P\begin{pmatrix} u \\ \varepsilon \end{pmatrix} &= (2\pi)^{-\frac{n+q}{2}} \begin{vmatrix} \mathbf{\Omega_{ST}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{vmatrix}^{-1} \exp\left[-\frac{1}{2}\begin{pmatrix} u \\ \varepsilon \end{pmatrix}^T \begin{pmatrix} \mathbf{\Omega_{ST}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix}^{-1} \begin{pmatrix} u \\ \varepsilon \end{pmatrix}\right] \\ &= (2\pi)^{-\frac{n+q}{2}} \begin{vmatrix} \mathbf{\Omega_{ST}} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{vmatrix}^{-1} \exp\left[-\frac{1}{2}\begin{pmatrix} u \\ \varepsilon \end{pmatrix}^T \begin{pmatrix} \mathbf{\Omega_{ST}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} \end{pmatrix} \begin{pmatrix} u \\ \varepsilon \end{pmatrix}\right], \end{aligned}$$

where q is the number of elements in the random vector u and the dimension of u. If $\mathbf{\Omega_{ST}}$ and $\mathbf{R}$ are known, maximise $P\binom{u}{\varepsilon}$ is equivalent to minimise

$$Q(u,\varepsilon) = \begin{pmatrix} u \\ \varepsilon \end{pmatrix}^T \begin{pmatrix} \mathbf{\Omega_{ST}}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}^{-1} \end{pmatrix} \begin{pmatrix} u \\ \varepsilon \end{pmatrix} = u^T\mathbf{\Omega_{ST}}^{-1}u + \varepsilon^T\mathbf{R}^{-1}\varepsilon.$$

Note that $\varepsilon = \varepsilon(\beta, u) = \mathbf{Y} - \mathbf{X}\beta - \mathbf{Z}u$. Using results from matrix calculus, we have

$$\frac{\partial\varepsilon(\beta,u)}{\partial\beta} = -\mathbf{X}^T; \frac{\partial\varepsilon(\beta,u)}{\partial u} = \mathbf{Z}^T.$$

Therefore,

$$\frac{\partial Q(u,\varepsilon)}{\partial\beta} = \frac{\partial\left(u^T\mathbf{\Omega_{ST}}^{-1}u\right)}{\partial\beta} + \frac{\partial\varepsilon}{\partial\beta}\frac{\partial\left(\varepsilon^T\mathbf{R}^{-1}\varepsilon\right)}{\partial\varepsilon} \quad = -2\mathbf{X}^T\mathbf{R}^{-1}\varepsilon.$$

Similarly, we have

$$\frac{\partial Q(u,\varepsilon)}{\partial u} = \frac{\partial\left(u^T\mathbf{\Omega_{ST}}^{-1}u\right)}{\partial\boldsymbol{u}} + \frac{\partial\varepsilon}{\partial u}\frac{\partial\left(\varepsilon^T\mathbf{R}^{-1}\varepsilon\right)}{\partial\varepsilon} \quad = 2\mathbf{\Omega_{ST}}^{-1}u - 2\mathbf{Z}^T\mathbf{R}^{-1}\varepsilon.$$

Setting $\frac{\partial Q}{\partial\beta}$ and $\frac{\partial Q}{\partial u}$ to be 0 and replacing ε with $\varepsilon = \mathbf{Y} - \mathbf{X}\beta - \mathbf{Z}u$, we have

$$\frac{\partial Q}{\partial\beta} = 0 \Leftrightarrow \mathbf{X}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta} + \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z}\hat{u} = \mathbf{X}^T R^{-1}\mathbf{Y}. \tag{2.9}$$

$$\frac{\partial Q}{\partial u} = \mathbf{0} \Leftrightarrow \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta} + \left(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1}\right)\hat{u} = \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y}. \tag{2.10}$$

Organizing Equations (2.9) and (2.10), we have Henderson's mixed model equations in matrix form:

$$\begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Y} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y} \end{bmatrix}.$$

□

The solution of Henderson's mixed model equations (MME) can be written as

$$\hat{\beta} = (\mathbf{X}'(\mathbf{Z}\mathbf{\Omega_{ST}}\mathbf{Z}^T + \mathbf{R})^{-1}\mathbf{X})^{-1}\mathbf{X}'(\mathbf{Z}\mathbf{\Omega_{ST}}\mathbf{Z}^T + \mathbf{R})^{-1}\mathbf{Y}, \tag{2.11}$$

$$\hat{u} = \mathbf{\Omega_{ST}}\mathbf{Z}'(\mathbf{Z}\mathbf{\Omega_{ST}}\mathbf{Z}^T + \mathbf{R})^{-1}(\mathbf{Y} - \mathbf{X}\hat{\beta}). \tag{2.12}$$

We now show Equations (2.11) and (2.12) are solution of Equation (2.7).

Proof. Equation (2.7) is equivalent to

$$\mathbf{X}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta} + \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z}\hat{u} = \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Y}. \tag{2.13}$$

$$\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta} + (\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1})\hat{u} = \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y}. \tag{2.14}$$

$\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z}$ is positive definite and invertible because $\mathbf{Z}$ has independent columns and $\mathbf{R}$ is positive definite. Then, Equation (2.14) can be transformed into Equation (2.15)

$$(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta} + \hat{u} = (\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y}. \tag{2.15}$$

Left multiply $\mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z}$ in both sides and then we get

$$\mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta} + \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z}\hat{u} = \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y}. \tag{2.16}$$

Combining Equations (2.13) and (2.16), we can eliminate $\hat{u}$:

$$\mathbf{X}^T(\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1})\mathbf{X}\hat{\beta} = \mathbf{X}^T(\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1})\mathbf{Y}. \tag{2.17}$$

We define $\mathbf{W} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{ST}}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}$ and rewrite Equation (2.17) as

$$\mathbf{X}^T\mathbf{W}\mathbf{X}\hat{\beta} = \mathbf{X}^T\mathbf{W}\mathbf{Y}, \tag{2.18}$$

then

$$\hat{\beta} = (\mathbf{X}^T\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{W}\mathbf{Y}. \tag{2.19}$$

The only thing left is to prove $\mathbf{W} = \mathbf{V}^{-1}$ where $\mathbf{V} = \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T + \mathbf{R}$:

$$\begin{aligned}
\mathbf{VW} &= (\mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T + \mathbf{R})(\mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}) \\
&= \mathbf{I} + \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1} - \mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1} - \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1} \\
&= \mathbf{I} + \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1} - \mathbf{Z}((\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1} + \boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1})\mathbf{Z}^T\mathbf{R}^{-1} \\
&= \mathbf{I} + \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1} - \mathbf{Z}(\mathbf{I} + \boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z})(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1})\mathbf{Z}^T\mathbf{R}^{-1} \\
&= \mathbf{I} + \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1} - \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}(\boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1} + \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z})((\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1})\mathbf{Z}^T\mathbf{R}^{-1} \\
&= \mathbf{I} + \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1} - \mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{R}^{-1} \\
&= \mathbf{I}.
\end{aligned}$$

So we can replace $\mathbf{W}$ with $\mathbf{V}^{-1} = (\mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T + \mathbf{R})^{-1}$ in Equation (2.19) which becomes

$$\hat{\beta} = (\mathbf{X}'(\mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T + \mathbf{R})^{-1}\mathbf{X})^{-1}\mathbf{X}'(\mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T + \mathbf{R})^{-1}\mathbf{Y}.$$

Now we finished the proof for $\hat{\beta}$ and move on to $\hat{u}$. By subtracting $\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta}$ in both sides of Equation (2.13), we get

$$(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})\hat{u} = \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y} - \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta}. \tag{2.20}$$

Then,

$$\begin{aligned}
\hat{u} &= (\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y} - \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X}\hat{\beta}) \\
&= (\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}(\mathbf{Y} - \mathbf{X}\hat{\beta}) \\
&= (\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{V}\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\hat{\beta}) \\
&= (\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}\mathbf{Z}^T\mathbf{R}^{-1}(\mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T + \mathbf{R})\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\hat{\beta}) \\
&= (\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T + \mathbf{Z}^T)\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\hat{\beta}) \\
&= (\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})^{-1}(\mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Omega}_{\mathbf{ST}}{}^{-1})\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\hat{\beta}) \\
&= \boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T\mathbf{V}^{-1}(\mathbf{Y} - \mathbf{X}\hat{\beta}) \\
&= \boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T(\mathbf{Z}\boldsymbol{\Omega}_{\mathbf{ST}}\mathbf{Z}^T + \mathbf{R})^{-1}(\mathbf{Y} - \mathbf{X}\hat{\beta}).
\end{aligned}$$

□

BLUPs have the smallest mean squared error of prediction among all linear unbiased predictors if the parameters in variance-covariance matrix are known. However, in practice, matrices $\boldsymbol{\Omega}_{\mathbf{ST}}$ and $\mathbf{R}$ are usually unknown. Therefore, specification of different choices of covariance matrices usually comes prior to estimation of β and u. We will cover it in Chapter 3.

2.3. Motivation of Mixed Effects Models

Simple linear regression (Equation 2.1) is not the best choice to model nested data due to within-field correlation, which violates the independence of errors assumption. Recall the example used at the beginning of Chapter 2, there are two levels of variables because plants (level 1) are nested in fields (level 2). Plant observations are not independent, as within a given field plants are more similar. Figure 2.2 shows plants as dots within larger circles representing fields.

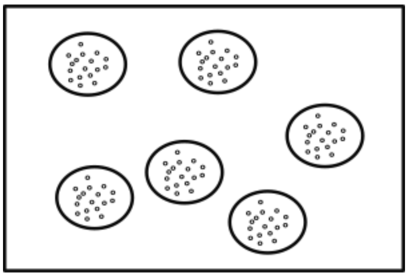


Figure 2.2: Plants in different fields

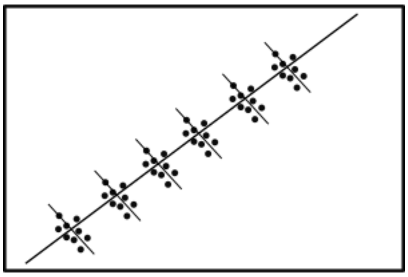


Figure 2.3: The model with the group variable

The within-field correlation leads to incorrect standard error estimates and, consequently, errors of statistical inferences. Underestimated standard errors cause an overestimation of test statistics, leading to inappropriate statistical significance and increased Type I error rate [8].

Ignoring the nested data structure can lead to underestimating standard errors and missing key relationships at each data level as well. We may miss important variables at the field level that help to explain difference at plant level because of not including information about the field. Thus, we applied an incorrect model to understand the outcome variable [9].

Figure 2.3 shows plants from six fields in a scatter plot of predictor versus outcome. Within fields, the relation is negative, but positive between fields. This is called Simpson's paradox. Including the group variable allows us to explore these important effects also avoid Simpson's paradox [9].

One method for handling nested data is aggregation, which averages plant data within a field to produce independent field data. While this approach gives consistent effect estimates and standard errors, it doesn't utilize all individual-level data. Another method is to run separate linear regressions for each field. This approach generates multiple models but doesn't leverage data from other fields, leading to larger variance in estimates [8, 9].

Mixed effects models can be thought of as a trade off between these two methods. The regression by group has many estimates and lots of data, but is noisy. The aggregated regression is less noisy, but may lose important differences by averaging all samples within each plot. Mixed effects models are somewhere in between caring about getting standard errors corrected for dependence in the data [8, 9].

3

Mixed Effects Models in Breeding

Mixed effects models are widely used in breeding because every phenotypic observation is determined by environmental and genetic factors: Phenotype = Environment + Genotype. The environment includes non-genetic influences on performance. Breeding programs also use observations of related individuals to infer their genotypic values. Mixed effects models integrate data from environment and related individuals to improve breeding value estimates. Breeding values are average effects of genes that are transmitted by a parent to an offspring [13, 14].

3.1. The Breeding Value Estimation

In general, the interest of breeding is on prediction of breeding values which are treated as random effect coefficients u in mixed effects models, and estimation of variance components [15]. The most straightforward information for estimating breeding values is the phenotype. Additionally, we can also use phenotypic data from relatives, including the father, mother, siblings, and progeny. If pedigree information is available, the BLUP method can incorporate data from relatives effectively [15, 16].

The mixed effects model allows efficient estimation of genetic parameters such as breeding values and variance components [17]. Breeding values are predicted using the following mixed effects model:

$$\mathbf{Y} = \mathbf{X}\beta + \mathbf{Z}u + \varepsilon,$$

where **Y** is a vector of phenotypic observations; **X** is a vector of fixed effects (Environment); **Z** is a vector of random effects (Genotype); ε is a vector of errors. Genetic evaluation by BLUP is heavily dependent on the genetic variance covariance matrix, both for higher accuracy and for unbiased results.

We assumed $u \sim \mathcal{N}(\mathbf{0}, \mathbf{\Omega_{ST}})$ in Assumption 2.1 and now assume $\mathbf{\Omega_{ST}} = \mathbf{\Omega_{ST0}}\sigma_u^2$, with $\mathbf{\Omega_{ST0}}$ as the additive genetic relationship matrix and σ_u^2 as the additive genetic variance. $\mathbf{\Omega_{ST0}}$ is about similarity between individuals. Off-diagonal elements represent the proportion of genes shared by two individuals. The diagonal element is the degree of inbreeding of the individual. Additive genetic variance σ_u^2 involves the inheritance of a particular allele from parents and this allele's independent effect on the specific phenotype, which will cause the phenotype deviation from the mean phenotype [17].

Assumption 3.1. $\mathbf{\Omega_{ST}} = \mathbf{\Omega_{ST0}}\sigma_u^2$, *where* $\mathbf{\Omega_{ST0}}$ *is the additive genetic relationship matrix in* $N \times N$, *similarity between individuals, and* σ_u^2 *is the additive genetic variance.*

We will cover how to specify $\mathbf{\Omega_{ST0}}$ by pedigree relationship matrix **A**, genomic relationship matrix **G** and combined relationship matrix **H** in the rest of this chapter. Matrix **A** is the relationship matrix from pedigree. Matrix **G** is the relationship matrix from genomic data. Matrix **H** combines information from matrices **A** and **G**.

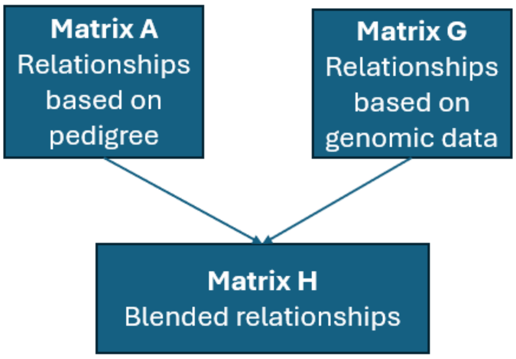

Matrix A
Relationships based on pedigree
Matrix G
Relationships based on genomic data
Matrix H
Blended relationships

Figure 3.1: The relationships among relationship matrices

3.2. Pedigree Relationship Matrix A & ABLUP

The pedigree relationship matrix **A** is used in breeding to get information from relatives in pedigree. It is a measure of the expected relationship among relatives. The assumption is that each parent should provide a random sample out of every two alleles. One allele would be randomly sampled from the two to create a new offspring. Matrix **A** is constrained by pedigree depth, completeness, and recording accuracy [18, 19].

Steps to calculate **A**:

- Step 1: Order the list of individuals chronologically so that parents precede offspring.
- Step 2: Calculate **A** row by row and recursively by

$$\mathbf{A}_{i,j} = \frac{1}{2}(A_{i,father(j)} + A_{i,mother(j)}),$$

$$\mathbf{A}_{i,i} = 1 + C_{inbreeding} A_{father(i),mother(i)}, \tag{3.1}$$

where $father(i)$ and $mother(i)$ are the father and mother of individual i. $C_{inbreeding}$ is a positive constant.

Assumption 3.2. *In pedigree there are individuals whose parents are unknown. The unknown parents are considered unrelated and non-inbred.*

Off-diagonal element $\mathbf{A}_{i,j}$ is about the expected similarity between individuals i and j. This is the mean of the similarity between individual i and the parents of individual j. Diagonal element $\mathbf{A}_{i,i}$ is the expected degree of inbreeding or the relationship between one and itself.

In equation (3.1), 1 is the relationship between one and itself without inbreeding. $C_{inbreeding} A_{father(i),mother(i)}$ is inbreeding coefficient. In practice, everyone uses $1/2$ for $C_{inbreeding}$ but it is unclear why this has been chosen. In genetics, the inbreeding coefficient indicates the chance that a individual inherits the same allele from both parents due to their genetic relatedness. In other words: it measures the probability of an individual inheriting the same allele from both parents due to their common ancestry [20].

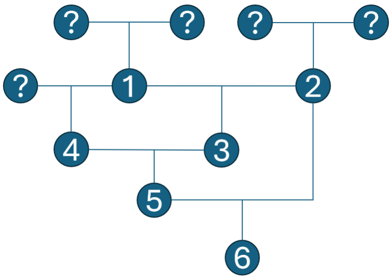


Figure 3.2: Example of pedigree chart

Individual	Father	Mother
1	0	0
2	0	0
3	1	2
4	1	0
5	4	3
6	5	2

Table 3.1: The ordered pedigree of Figure 3.2
The first column is the individual's number. The second and third columns are the individual's father and mother, respectively.

Figure 3.2 is an example of pedigree chart for the calculation of matrix $\mathbf{A}$ [21]. In step one, individuals from pedigree are sorted from oldest (top) to youngest (bottom). Table 3.1 displays ordered pedigree of Figure 3.2. The first column of Table 3.1 is the individual's number. The second and third columns are the individual's father and mother, respectively.

In step two, we calculate the diagonal and off-diagonal elements. For instance, the diagonal element for individual 6 in Figure 3.2 is $a_{6,6} = 1 + 0.5(a_{5,2}) = 1 + 0.5 \times 0.25 = 1.125$ because its parents are individuals 5 and 2. The expected relationship between individuals 1 and 6 is given by $a_{1,6} = 0.5(a_{1,5} + a_{1,2}) = 0.5 \times (0.5 + 0) = 0.25$, as animal 5 is the father and animal 2 is the mother of 6, making the relationship between 1 and 6 the average of 1's relationships with 5 and 2. The pedigree relationship matrix for all individuals in Figure 3.2 is displayed in Table 3.2.

Plant	1	2	3	4	5	6
1	1	0	0.5	0.5	0.5	0.25
2	0	1	0.5	0	0.25	0.625
3	0.5	0.5	1	0.25	0.625	0.5625
4	0.5	0	0.25	1	0.625	0.3125
5	0.5	0.25	0.625	0.625	1.125	0.6875
6	0.25	0.625	0.5625	0.3125	0.6875	1.125

Table 3.2: The Matrix A of individuals in Figure 3.2

Matrix $\mathbf{A}$ is called "relationship matrix" but it is a variance covariance matrix meaning that on the diagonal elements are variances, and the off-diagonal elements are covariances. To obtain the relationships one would divide the off-diagonal elements by the square roots of the product of the corresponding diagonals: $\mathbf{diag(A)}^{-1/2}\mathbf{Adiag(A)}^{-1/2}$. This reduces a variance covariance matrix to a correlation matrix. Thus, we also call matrix $\mathbf{A}$ the numerator relationship matrix [18].

ABLUP is the conventional BLUP method using pedigree relationship matrix $\mathbf{A}$ as $\mathbf{\Omega_{ST0}}$ to estimate breeding values. Therefore, the distribution of random effect coefficients u (Assumption 2.1) can be rewritten as $\mathcal{N}(\mathbf{0}, \mathbf{A}\boldsymbol{\sigma}_{\boldsymbol{u}}^{\mathbf{2}})$. The distribution of $\mathbf{Y} \mid \mathbf{X}, \mathbf{Z}$ can be expressed as: $\mathcal{N}(\mathbf{X}\beta, \mathbf{Z}\mathbf{A}\boldsymbol{\sigma}_{\boldsymbol{u}}^{\mathbf{2}}\mathbf{Z}^T + \mathbf{R})$. If the residual vector ε is assumed to satisfy $\varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}\sigma_\varepsilon^2)$ where σ_ε^2 is the residual variance, Equation (2.7) can be expressed as:

$$\begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{A}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Y} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y} \end{bmatrix}.$$

ABLUP is given by: $\hat{u} = (\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{A}^{-1})^{-1}\mathbf{Z}^T(\mathbf{Y} - \mathbf{X}\hat{\beta})$ with $\lambda = \sigma_\varepsilon^2/\sigma_u^2$.

Assumption 3.3. *The distribution of error terms ε is a special case of Assumption 2.2, where $\mathbf{R} = \mathbf{I}\sigma_\varepsilon^2$:*
$\varepsilon \sim \mathcal{N}(0, \mathbf{I}\sigma_\varepsilon^2)$.

3.3. Genomic Relationship Matrix G & GBLUP

The matrix $\mathbf{A}$ relies on pedigree information to estimate the relationships between relatives. The matrix $\mathbf{G}$ uses DNA information by a large number (10^4) of SNP markers (single nucleotide polymorphism). It is not feasible to measure all DNA information which gives us the proportion of chromosome segments shared between individuals. Genomic relationships in matrix $\mathbf{G}$ provide accurate estimate of relationships. This is because high-density genotyping can identify genes that are identical in state, which may be inherited from common ancestors not documented in the pedigree [22].

An individual's breeding value consists of half from the father, half from the mother, and the Mendelian sampling term. The Mendelian sampling represents individual's difference from the average of its parents' breeding values and is due to the random sample of genes and chromosomes that the progeny inherited [23]. Therefore, $\mathbf{G}_{i,j}$ is a random variable with $\mathbb{E}[\mathbf{G}_{i,j}] = \mathbf{A}_{i,j}$. Relationships in matrix $\mathbf{G}$ can deviate from the expected relationship given in matrix **A** because matrix **A** only includes the breeding value of father and mother. For example, the relationship between two full siblings in **G** may range from 0.4 to 0.6 because of Mendelian sampling instead of 0.5 given in matrix **A** [23].

To calculate matrix **G**, we first introduce the marker genotype matrix **M**. Matrix **M** indicates the genetic markers inherited by each individual. A genetic marker is a gene or DNA sequence with a known location on a chromosome that can be used to identify individuals or species [24]. The dimensions of matrix **M** are defined by the number of individuals N and the number of markers $N_{markers}$ [19, 25]. If B and b are alleles, the elements in matrix **M** are defined as

$$\mathbf{M}_{i,j} = \begin{cases} 0 \text{ if the individual is homozygous for the first allele, BB} \\ 1 \text{ if it is heterozygous, Bb} \\ 2 \text{ if it is homozygous for the second allele, bb.} \end{cases}$$

Secondly, we introduce matrix **P** which will be used to center matrix **G**. Centering measures genetic variances and covariances as deviations from the average genotype [19]. In genetics, a locus (plural: loci) is a specific, fixed position on a chromosome where a particular gene or genetic marker is located [26]. Let the estimated minor allele frequency at locus j be p_j with $j \in \{1, 2, \ldots N_{markers}\}$. The minor allele is the second most common allele occurs also known as the second allele. In this example the minor allele is b. **P** is a $N \times N_{markers}$ matrix defined by $\mathbf{P} = (\mathbf{P}_{ij})$ with $\mathbf{P}_{ij} = 2(p_j - 0.5)$ where $i \in \{1, 2, \ldots N\}$. Then the formulation of **G** is

$$\mathbf{G} = \frac{(\mathbf{M} - \mathbf{P})(\mathbf{M} - \mathbf{P})^T}{C_G} = \frac{\mathbf{K}\mathbf{K}^T}{C_G}. \tag{3.2}$$

In practice, everyone uses $2\sum_{i=1}^{N_{locus}} p_i(1 - p_i)$ as C_G but it is unclear why this has been chosen. The genomic inbreeding coefficient for individual i is $\mathbf{G}_{i,i} - 1$. The genomic relationships between individuals i and j are obtained by $\mathbf{G}_{i,j}/\sqrt{\mathbf{G}_{i,i}\mathbf{G}_{j,j}}$ [19, 25].

Genomic BLUP (GBLUP) is a way to utilize genotypes to estimate breeding values. Genomic relationship matrix **G** is used instead of pedigree relationship matrix **A**. By replacing the relationship matrix from **A** to **G**, The distribution of u changes to $\mathcal{N}(\mathbf{0}, \mathbf{G}\sigma_u^2)$ from $\mathcal{N}(\mathbf{0}, \mathbf{A}\sigma_u^2)$ and the distribution of $\mathbf{Y} \mid \mathbf{X}, \mathbf{Z}$ is $\mathcal{N}(\mathbf{X}\beta, \mathbf{Z}\mathbf{G}\sigma_u^2\mathbf{Z}^T + \mathbf{R})$. The MME Equation (2.7) can be expressed as:

$$\begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Y} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y} \end{bmatrix}.$$

GBLUP is given by: $\hat{u} = (\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{G}^{-1})^{-1}\mathbf{Z}^T(\mathbf{Y} - \mathbf{X}\hat{\beta})$ with $\lambda = \sigma_\varepsilon^2/\sigma_u^2$.

Assumption 3.4. *The distribution of random effect coefficients u is $\mathcal{N}(\mathbf{0}, \mathbf{G}\sigma_u^2)$ in GBLUP.*

3.4. Single-Step GBLUP

GBLUP faces a challenge: phenotypes are available on thousands or millions of individuals, yet there are genotypes on a select subset of those because genotyping is quite expensive on a population basis for all individuals. Single-Step GBLUP (ssGBLUP) addresses it by using all phenotypes with a combined relationship matrix **H** [27]. Matrix **H** combines information from matrices **A** and **G**, serving as an estimator of relationships using both pedigree and genomic information. Calculation of matrix **G** is important in ssGBLUP. Matrices **A** and **G** can be divided into four parts:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}, \mathbf{G} = \begin{bmatrix} \mathbf{G}_{11} & \mathbf{G}_{12} \\ \mathbf{G}_{21} & \mathbf{G}_{22} \end{bmatrix},$$

where subscripts 1 and 2 represent ungenotyped and genotyped individuals. While **G** is partially observed with only $\mathbf{G}_{22}$ observed. The simplest way to combine matrix **A** and matrix **G** is to replace $\mathbf{A}_{22}$ with matrix **G**:

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{G}_{22} \end{bmatrix}. \tag{3.3}$$

It is simple to use but can be improved because there may be connections between genotyped and non-genotyped individuals through pedigree.

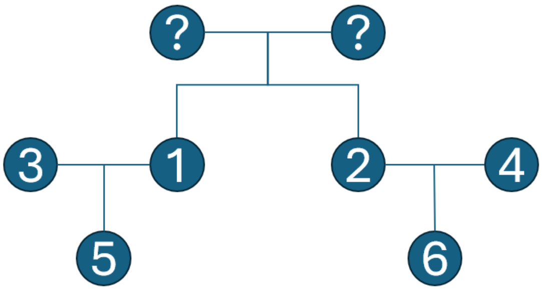


Figure 3.3: The relationships of non-genotyped individuals may also change because of $\mathbf{G}_{22}$.

Matrix $\mathbf{G}_{22}$ may modify relationships in the ancestors and descendants of genotyped individuals, as demonstrated by the pedigree in Figure 3.3. Assume two full-siblings, 1 and 2, are genotyped with genomic relationship 0.4 and individuals 5 and 6 are non-genotyped. By using Matrix (3.3), we get the estimated relationship between 5 and 6 is 0.125. 1 and 2 are full siblings so they share 50% gene. 5 and 6 get 50% of the genes from 1 and 2 respectively. So 5 and 6 share 0.5^2 gene form 1 and 2. Since they are only related through 1 and 2 so 5 and 6 share $0.5^3 = 0.125$ of the genes. Actually, we can get a better estimate by $0.4 \times 0.5 \times 0.5 = 0.1$ because we know the genetic relationship between 1 and 2 is 0.4. Therefore, adding information from matrix $\mathbf{G}_{22}$ to matrix **A** alters the relationships among genotyped individuals, and may also affect non-genotyped ones [27].

Assumption 3.5. *In the following of this section, it is assumed that $\sigma_u^2 = 1$.*

We are going to show how to improve Matrix (3.3) to estimate the relationships which means we need to compute $\text{Cov}(u \mid \mathbf{G_{22}}, \mathbf{A})$. The following method was developed by Legarra et al. [27].

Theorem 1. *We assume $u \mid \mathbf{A} \sim \mathcal{N}(0, \mathbf{A})$ and $u_2 \mid \mathbf{G} \sim \mathcal{N}(0, \mathbf{G_{22}})$. Then*

$$Cov(u \mid \mathbf{G_{22}}, \mathbf{A}) := \mathbf{H} = \mathbf{A} + \begin{bmatrix} \mathbf{A_{12}A_{22}^{-1}(G - A_{22})A_{22}}^{-1}\mathbf{A_{21}} & \mathbf{A_{12}A_{22}}^{-1}\mathbf{(G - A_{22})} \\ \mathbf{(G - A_{22})A_{22}^{-1}A_{21}} & \mathbf{G - A_{22}} \end{bmatrix}.$$

Proof. According to the conditional distribution of multivariate normal distribution in Appendix B, the distribution of breeding values of ungenotyped individuals, conditioned on breeding values of genotyped individuals, is:

$$f(u_1 \mid u_2) = \mathcal{N}(\mathbf{A_{12}A_{22}}^{-1}u_2, \mathbf{A_{11}} - \mathbf{A_{11}A_{22}}^{-1}\mathbf{A_{21}}),$$

or:

$$u_1 = \mathbb{E}(u_1 \mid u_2) + \epsilon = \mathbf{A_{12}A_{22}}^{-1}u_2 + \epsilon,$$

$$\epsilon \sim \mathcal{N}(0, \mathbf{A_{11}} - \mathbf{A_{11}A_{22}}^{-1}\mathbf{A_{21}}).$$

It can be expressed as a regression equation:

$$u_1 = \mathbf{A_{12}A_{22}}^{-1}u_2 + \epsilon.$$

Recall that $\text{Var}(u_2) = \mathbf{G}$. So that

$$\begin{aligned} \text{Var}(u_1) &= \mathbf{A_{12}A_{22}}^{-1}\mathbf{G}\mathbf{A_{22}}^{-1}\mathbf{A_{21}} + \mathbf{A_{11}} - \mathbf{A_{12}A_{22}}^{-1}\mathbf{A_{21}} \\ &= \mathbf{A_{11}} + \mathbf{A_{12}A_{22}^{-1}(G - A_{22})A_{22}}^{-1}\mathbf{A_{21}}, \end{aligned}$$

and

$$\text{Cov}(u_1, u_2) = \mathbf{A_{12}A_{22}}^{-1}\mathbf{G},$$

Now we can write matrix **H** as:

$$\begin{aligned}
\mathbf{H} = \begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} &= \begin{bmatrix} \mathrm{Var}(u_1) & \mathrm{Cov}(u_2, u_1) \\ \mathrm{Cov}(u_1, u_2) & \mathrm{Var}(u_2) \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{A}_{11} + \mathbf{A}_{12}{\mathbf{A}_{22}}^{-1}(\mathbf{G} - \mathbf{A}_{22}){\mathbf{A}_{22}}^{-1}\mathbf{A}_{21} & \mathbf{A}_{12}{\mathbf{A}_{22}}^{-1}\mathbf{G} \\ \mathbf{G}{\mathbf{A}_{22}}^{-1}\mathbf{A}_{21} & \mathbf{G} \end{bmatrix} \\
&= \mathbf{A} + \begin{bmatrix} \mathbf{A}_{12}\mathbf{A}_{22}^{-1}(\mathbf{G} - \mathbf{A}_{22}){\mathbf{A}_{22}}^{-1}\mathbf{A}_{21} & \mathbf{A}_{12}{\mathbf{A}_{22}}^{-1}(\mathbf{G} - \mathbf{A}_{22}) \\ (\mathbf{G} - \mathbf{A}_{22})\mathbf{A}_{22}^{-1}\mathbf{A}_{21} & \mathbf{G} - \mathbf{A}_{22} \end{bmatrix}.
\end{aligned}$$

□

Matrix **H** is a variation of the pedigree relationships matrix **A** to include genomic relationships in **G**. Notice how **G** comes into the $\mathbf{H}_{11}$, $\mathbf{H}_{12}$, and $\mathbf{H}_{21}$ parts in Theorem 1. This implies that genomic relationships would change the connections among non-genotyped individuals, as well as between genotyped and non-genotyped individuals [28]. Two unrelated individuals in **A** will appear as related in **H** if their descendants are related in **G**. Accordingly, two descendants of individuals that are related in **G** will be related in **H**, even if they are not related in matrix **A**.

There are other different ways to understand the matrix **H**: in this matrix, genomic information is conveyed through the pedigree of all individuals. It is a projection of **G** onto the remaining individuals. Additionally, it is also a Bayesian update of matrix **A** based on new information from genotypes [27, 28].

Besides, genotyped individuals without phenotypic data should be excluded in matrix **A**. While they cannot be excluded from matrix **H**. This is because, unless both parents are genotyped, these individuals can still influence the pedigree relationships of other individuals, particularly their parents. We can use the information to get the better estimation of the relationships between individuals.

The pedigree chart in Figure 3.4 is an example to show it. Consider two half-siblings 4 and 5 born to one father and two unrelated, non-genotyped mothers 2 and 3. If individuals 4 and 5 are nearly identical, matrix **H** will capture this information, resulting in the non-genotyped mothers 2 and 3 being treated as related or even identical within matrix **H** [28].

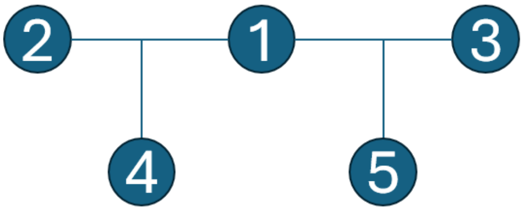


Figure 3.4: Genotyped offsprings influence relationship between non-genotyped parents.

In Theorem 1, we assumed that $u \mid \mathbf{A} \sim \mathcal{N}(0, \mathbf{A})$. Existing researches for joint analysis of genotyped and ungenotyped individuals also have this assumption [29, 30]. Strictly speaking, this is not coherent in our framework because $f(u \mid \mathbf{A}, \mathbf{K_2})$ is a mixture of multivariate normal densities [15]. The matrix $\mathbf{K}$ was defined in Equation (3.2). The part of the matrix $\mathbf{K}$ used to compute $\mathbf{G}_{11}$ and $\mathbf{G}_{22}$ are defined as $\mathbf{K}_1$ and $\mathbf{K}_2$ but $\mathbf{K}_1$ is not observed in practice. Conditional distribution $f(u \mid \cdots)$ can be normal distribution or mixture of normal distributions. It depends on the set of variables we are conditioning on as shows in the following theorem.

Theorem 2. *Assume $u \mid \mathbf{G} \sim \mathcal{N}(0, \mathbf{G})$ and $\mathbb{E}(\mathbf{G}) = \mathbf{A}$. Then $f(u \mid \mathbf{K_1}, \mathbf{K_2})$ is a multivariate normal distribution and $f(u \mid \mathbf{A}, \mathbf{K_2})$ a mixture of multivariate normal distributions.*

Proof. When all individuals are genotyped, the marginal distribution of breeding values u is

$$f(u \mid \mathbf{K}) = \mathcal{N}(0, \mathbf{G}) = \frac{1}{C_G}\mathcal{N}(0, \begin{bmatrix} \mathbf{K_1}\mathbf{K_1}^T & \mathbf{K_1}\mathbf{K_2}^T \\ \mathbf{K_2}\mathbf{K_1}^T & \mathbf{K_2}\mathbf{K_2}^T \end{bmatrix}),$$

which is a multivariate normal distribution. When some individuals are not genotyped, we need to derive the joint density of genetic values given $\mathbf{A}$ and $\mathbf{K_2}$:

$$\begin{aligned} f(u \mid \mathbf{A}, \mathbf{K_2}) &= \int f(u_1, u_2, \mathbf{K_1} \mid \mathbf{K_2}, \mathbf{A}) d\mathbf{K_1} \\ &= \int f(u_1, u_2, \mid \mathbf{K_1}, \mathbf{K_2}) f(\mathbf{K_1} \mid \mathbf{K_2}, \mathbf{A}) d\mathbf{K_1} \\ &= \int \mathcal{N}(0, \mathbf{G}) f(\mathbf{K_1} \mid \mathbf{K_2}, \mathbf{A}) d\mathbf{K_1}, \end{aligned}$$

which is a mixture of multivariate normal distributions. □

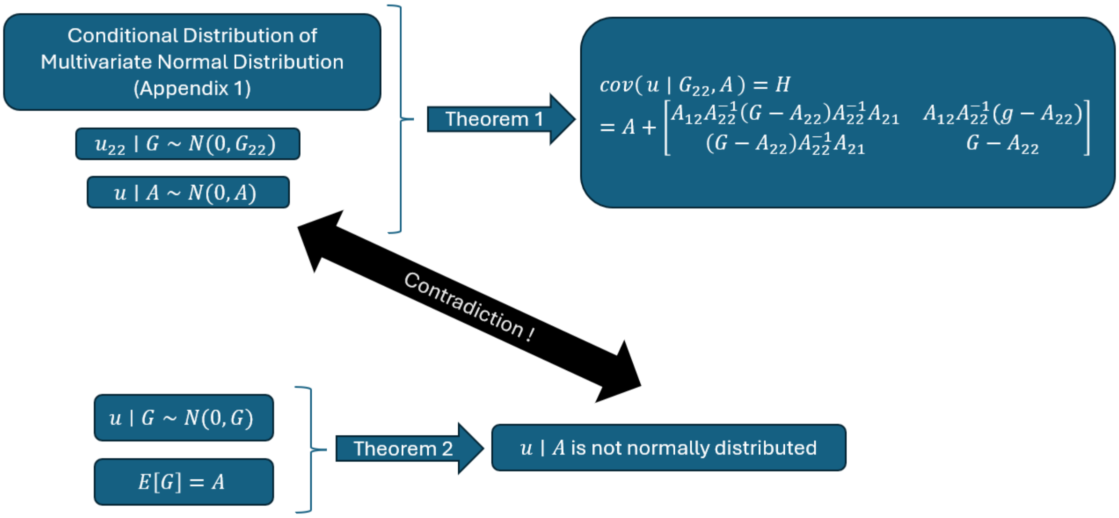

Conditional Distribution of Multivariate Normal Distribution (Appendix 1)
$u_{22} \mid G \sim N(0, G_{22})$
$u \mid A \sim N(0, A)$
Theorem 1
$cov(u \mid G_{22}, A) = H$
$= A + \begin{bmatrix} A_{12}A_{22}^{-1}(G - A_{22})A_{22}^{-1}A_{21} & A_{12}A_{22}^{-1}(g - A_{22}) \\ (G - A_{22})A_{22}^{-1}A_{21} & G - A_{22} \end{bmatrix}$
Contradiction !
$u \mid G \sim N(0, G)$
$E[G] = A$
Theorem 2
$u \mid A$ is not normally distributed

Figure 3.5: $u \mid \mathbf{A} \sim \mathcal{N}(0, \mathbf{A})$ is not coherent with assumptions in Theorem 2.

This is summarized in Figure 3.5. $u \mid \mathbf{A} \sim \mathcal{N}(0, \mathbf{A})$ is not coherent with assumptions in Theorem 2. But $u \mid \mathbf{A}$ may follow a distribution "close" to $\mathcal{N}(0, \mathbf{A})$, which justifies the use of $\mathbf{H}$ as an approximation of $Cov(u \mid \mathbf{G}_{22}, \mathbf{A})$. It is because we can not conclude $u \sim \mathcal{N}(0, \mathbf{A})$ from $u \mid \mathbf{G} \sim \mathcal{N}(0, \mathbf{G})$ and $\mathbb{E}(\mathbf{G}) = \mathbf{A}$. We are going to give a numeric example where $\mathbf{A}$ and $\mathbf{G}$ are scalars. Let $\mathbf{A} = 0.5$ and $\mathbf{G}$ is in Bernoulli distribution: $\mathbf{G} \sim Bern(1/2)$ so $\mathbb{E}(\mathbf{G}) = \mathbf{A}$. The true distribution of u is a mixture of normal distributions, $\frac{1}{2}\mathcal{N}(0,1) + \frac{1}{2}\mathcal{N}(0,3)$, while $\mathcal{N}(0, \mathbf{A}) = \mathcal{N}(0,2)$. Figure 3.6 presents the difference between the two distributions. Therefore, $\mathcal{N}(0, \mathbf{A})$ is not the real distribution of u but an approximation of it.

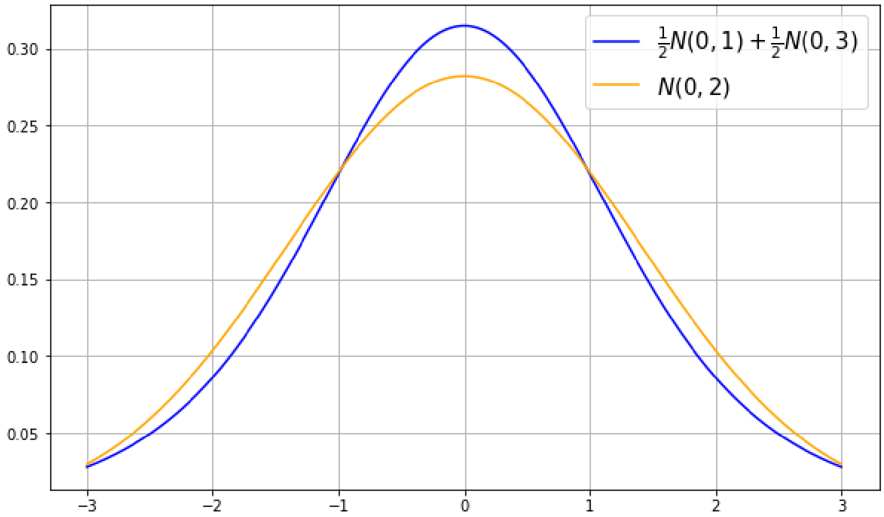


Figure 3.6: PDFs of the mixture distribution and normal distribution

3.5. Multi-Trait Single-Step GBLUP

We have worked on single-trait models so far. For multiple traits, genetic evaluation can be done either individually using single-trait models or simultaneously with multi-trait models [31]. Recall N is the number of individuals with p fixed effects and q random effects. We begin with an example using $p = 2$, $q = 0$, and $N = 3$. We choose $q = 0$ for simplicity as the fixed effects model extends easily to the mixed effects model. $\mathbf{Y}_i \in \mathbb{R}$ is the phenotype data of i^{th} individual. $\mathbf{X}_{il}$ is the i^{th} individual's l^{th} fixed effect value. β_l is the coefficient of l^{th} fixed effects. A single-trait model is expressed as following when $k = 1$:

$$\mathbf{Y}_1 = \mathbf{X}_{11}\beta_1 + \mathbf{X}_{12}\beta_2 + \varepsilon_1 \mathbf{Y}_2 = \mathbf{X}_{21}\beta_1 + \mathbf{X}_{22}\beta_2 + \varepsilon_2 \mathbf{Y}_3 = \mathbf{X}_{31}\beta_1 + \mathbf{X}_{32}\beta_2 + \varepsilon_3.$$

The single-trait model in matrix form is:

$$\begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \mathbf{Y}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \\ \mathbf{X}_{31} & \mathbf{X}_{32} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \end{bmatrix}.$$

Continuing the example with $k = 2$ traits, it is a multi-trait model, also known as a bi-trait model. $\mathbf{Y}_{ij}$ is phenotype data for i^{th} individual's j^{th} trait. $\mathbf{X}_{il}$ represents the value of i^{th} individual's l^{th} fixed effect, and β_{jl} is the coefficient of l^{th} fixed effects on j^{th} trait. The multi-trait model is

$$\begin{cases} \mathbf{Y}_{11} = \mathbf{X}_{11}\beta_{11} + \mathbf{X}_{12}\beta_{12} + \varepsilon_{11} \\ \mathbf{Y}_{21} = \mathbf{X}_{21}\beta_{11} + \mathbf{X}_{22}\beta_{12} + \varepsilon_{21} \\ \mathbf{Y}_{31} = \mathbf{X}_{31}\beta_{11} + \mathbf{X}_{32}\beta_{12} + \varepsilon_{31} \\ \mathbf{Y}_{12} = \mathbf{X}_{11}\beta_{21} + \mathbf{X}_{12}\beta_{22} + \varepsilon_{12} \\ \mathbf{Y}_{22} = \mathbf{X}_{21}\beta_{21} + \mathbf{X}_{22}\beta_{22} + \varepsilon_{22} \\ \mathbf{Y}_{32} = \mathbf{X}_{31}\beta_{21} + \mathbf{X}_{32}\beta_{22} + \varepsilon_{32} \end{cases}.$$

Joint trait analysis can use two forms of the multi-trait model: stacked and non-stacked, which differ in matrix arrangement. The stacked form is:

$$\begin{bmatrix} \mathbf{Y}_{11} \\ \mathbf{Y}_{21} \\ \mathbf{Y}_{31} \\ \mathbf{Y}_{12} \\ \mathbf{Y}_{22} \\ \mathbf{Y}_{32} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathbf{X}_{21} & \mathbf{X}_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathbf{X}_{31} & \mathbf{X}_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{X}_{11} & \mathbf{X}_{12} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{X}_{21} & \mathbf{X}_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{X}_{31} & \mathbf{X}_{32} \end{bmatrix} \begin{bmatrix} \beta_{11} \\ \beta_{12} \\ \beta_{11} \\ \beta_{12} \\ \beta_{11} \\ \beta_{12} \\ \beta_{21} \\ \beta_{22} \\ \beta_{21} \\ \beta_{22} \\ \beta_{21} \\ \beta_{22} \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{21} \\ \varepsilon_{31} \\ \varepsilon_{12} \\ \varepsilon_{22} \\ \varepsilon_{32} \end{bmatrix}.$$

The non stacked form is:

$$\begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} \\ \mathbf{Y}_{21} & \mathbf{Y}_{22} \\ \mathbf{Y}_{31} & \mathbf{Y}_{32} \end{bmatrix} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} \\ \mathbf{X}_{21} & \mathbf{X}_{22} \\ \mathbf{X}_{31} & \mathbf{X}_{32} \end{bmatrix} \begin{bmatrix} \beta_{11} & \beta_{21} \\ \beta_{12} & \beta_{22} \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} \\ \varepsilon_{21} & \varepsilon_{22} \\ \varepsilon_{31} & \varepsilon_{32} \end{bmatrix}.$$

Following examples, we define the general multi-trait model with k traits from the single-trait model. The single-trait model for each trait is:

$$\overbrace{\mathbf{Y}_j}^{\mathrm{N\times 1}} = \overbrace{\underbrace{\mathbf{X}}_{\mathrm{N\times p}} \quad \underbrace{\beta_j}_{\mathrm{p\times 1}}}^{\mathrm{N\times 1}} + \overbrace{\underbrace{\mathbf{Z}}_{\mathrm{N\times q}} \quad \underbrace{u_j}_{\mathrm{q\times 1}}}^{\mathrm{N\times 1}} + \overbrace{\varepsilon_j}^{\mathrm{N\times 1}}. \tag{3.4}$$

Assumption 3.6. *To simplify the presentation of multi-trait model, we assume each individual has either no observations or observations on all traits.*

The stacked multi-trait model is defined as

$$\overbrace{\mathbf{Y}}^{\text{Nk}\times 1} = \overbrace{\underbrace{\mathbf{X}}_{\text{Nk}\times\text{Nkp}} \quad \underbrace{\beta}_{\text{Nkp}\times 1}}^{\text{Nk}\times 1} + \overbrace{\underbrace{\mathbf{Z}}_{\text{Nk}\times\text{Nkq}} \quad \underbrace{u}_{\text{Nkq}\times 1}}^{\text{Nk}\times 1} + \overbrace{\varepsilon}^{\text{Nk}\times 1} \tag{3.5}$$

with

$$\tilde{\mathbf{X}} = \begin{bmatrix} \mathbf{X}_{11} & \cdots & \mathbf{X}_{1p} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \mathbf{X}_{21} & \cdots & \mathbf{X}_{2p} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & \mathbf{X}_{N1} & \cdots & \mathbf{X}_{Np} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \tilde{\mathbf{X}} & 0 & \cdots & 0 \\ 0 & \tilde{\mathbf{X}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{\mathbf{X}} \end{bmatrix},$$

$$\tilde{\mathbf{Z}} = \begin{bmatrix} \mathbf{Z}_{11} & \cdots & \mathbf{Z}_{1q} & 0 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \cdots & 0 & \mathbf{Z}_{21} & \cdots & \mathbf{Z}_{2q} & \cdots & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & \cdots & \mathbf{Z}_{N1} & \cdots & \mathbf{Z}_{Nq} \end{bmatrix}, \mathbf{Z} = \begin{bmatrix} \tilde{\mathbf{Z}} & 0 & \cdots & 0 \\ 0 & \tilde{\mathbf{Z}} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \tilde{\mathbf{Z}} \end{bmatrix},$$

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{11} \\ \vdots \\ \mathbf{Y}_{N1} \\ \mathbf{Y}_{12} \\ \vdots \\ \mathbf{Y}_{N2} \\ \vdots \\ \mathbf{Y}_{1k} \\ \vdots \\ \mathbf{Y}_{Nk} \end{bmatrix}, \tilde{\beta}_i = \begin{bmatrix} \beta_{i1} \\ \vdots \\ \beta_{ip} \end{bmatrix}, \beta = \begin{bmatrix} \tilde{\beta}_1 \\ \vdots \\ \tilde{\beta}_1 \\ \tilde{\beta}_2 \\ \vdots \\ \tilde{\beta}_2 \\ \vdots \\ \tilde{\beta}_k \\ \vdots \\ \tilde{\beta}_k \end{bmatrix}, \tilde{u}_i = \begin{bmatrix} u_{i1} \\ \vdots \\ u_{ip} \end{bmatrix}, u = \begin{bmatrix} \tilde{u}_1 \\ \vdots \\ \tilde{u}_1 \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_2 \\ \vdots \\ \tilde{u}_k \\ \vdots \\ \tilde{u}_k \end{bmatrix}, \varepsilon = \begin{bmatrix} \varepsilon_{11} \\ \vdots \\ \varepsilon_{N1} \\ \varepsilon_{12} \\ \vdots \\ \varepsilon_{N2} \\ \vdots \\ \varepsilon_{1k} \\ \vdots \\ \varepsilon_{Nk} \end{bmatrix}.$$

The non stacked multi-trait model is defined as

$$\overbrace{\mathbf{Y}}^{\text{N}\times\text{k}} = \overbrace{\underbrace{\mathbf{X}}_{\text{N}\times\text{p}} \quad \underbrace{\beta}_{\text{p}\times\text{k}}}^{\text{N}\times\text{k}} + \overbrace{\underbrace{\mathbf{Z}}_{\text{N}\times\text{q}} \quad \underbrace{u}_{\text{q}\times\text{k}}}^{\text{N}\times\text{k}} + \overbrace{\varepsilon}^{\text{N}\times\text{k}} \tag{3.6}$$

with

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{Y}_{12} & \cdots & \mathbf{Y}_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{Y}_{N1} & \mathbf{Y}_{N2} & \cdots & \mathbf{Y}_{Nk} \end{bmatrix}, \varepsilon = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \cdots & \varepsilon_{1k} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{N1} & \varepsilon_{N2} & \cdots & \varepsilon_{Nk} \end{bmatrix}, \mathbf{X} = \begin{bmatrix} \mathbf{X}_{11} & \mathbf{X}_{12} & \cdots & \mathbf{X}_{1p} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{N1} & \mathbf{X}_{N2} & \cdots & \mathbf{X}_{Np} \end{bmatrix},$$

$$\beta = \begin{bmatrix} \beta_{11} & \beta_{21} & \cdots & \beta_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{1p} & \beta_{2p} & \cdots & \beta_{kp} \end{bmatrix}, u = \begin{bmatrix} u_{11} & u_{21} & \cdots & u_{k1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{1q} & u_{2q} & \cdots & u_{kq} \end{bmatrix}.$$

Denote the Kronecker product by $\otimes$. If A is an $m \times n$ matrix and B a $p \times q$ matrix, their Kronecker product $A \otimes B$ forms a $pm \times qn$ block matrix:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

Assumption 3.7. *In stacked multi-trait model $u \sim \mathcal{N}(0, \mathbf{\Omega_{MT}} \otimes \mathbf{H})$, where*

$$\mathbf{\Omega_{MT}} = \begin{bmatrix} \sigma^2_{u_1} & \sigma_{u_{12}} & \cdots & \sigma_{u_{1k}} \\ \sigma_{u_{12}} & \sigma^2_{u_2} & \cdots & \sigma_{u_{2k}} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{u_{1k}} & \sigma_{u_{2k}} & \cdots & \sigma^2_{u_k} \end{bmatrix}$$

is the matrix of genetic covariance across traits and $\mathbf{H}$ *is the combined relationship matrix [32, 33].*

Assumption 3.8. *In stacked multi-trait model $\varepsilon \sim \mathcal{N}(0, \mathbf{R})$ and $\mathbf{R} = \mathbf{R_0} \otimes \mathbf{I_N}$, where*

$$\mathbf{R}_0 = \begin{bmatrix} \sigma^2_{e_1} & \sigma_{e_{12}} & \cdots & \sigma_{e_{1k}} \\ \sigma_{e_{12}} & \sigma^2_{e_2} & \cdots & \sigma_{e_{2k}} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{e_{1k}} & \sigma_{e_{2k}} & \cdots & \sigma^2_{e_k} \end{bmatrix}$$

is the matrix of residual covariance across traits and $\mathbf{I_N}$ *is the $N \times N$ identity matrix [32, 33].*

Replacing $\mathbf{\Omega_{ST}}^{-1}$ by $\mathbf{\Omega_{MT}}^{-1} \otimes \mathbf{H}^{-1}$ in Equation (2.7) leads to the MME of multi-trait ssGBLUP:

$$\begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Z} + \mathbf{\Omega_{MT}}^{-1} \otimes \mathbf{H}^{-1} \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} \mathbf{X}^T\mathbf{R}^{-1}\mathbf{Y} \\ \mathbf{Z}^T\mathbf{R}^{-1}\mathbf{Y} \end{bmatrix}.$$

3.6. Genetic Correlation ρ^{Gen}

Genetic correlation quantifies the genetic relationship between two traits by measuring the proportion of shared genetic variance, which in turn influences their phenotypic correlation. However, if the environmental correlation is strong enough in the opposite direction, the genetic correlation can have different direction with the phenotypic correlation [7]. Genetic correlation plays a key role in quantitative genetics and breeding for indirect selection, particularly when measuring traits directly is costly. If trait a is easily observed and genetically correlated to trait b, improving trait a can impact trait b [34].

In a quantitative genetic model, traits a and b are defined as the sum of genetic value g and residual value e, which simply means the difference between the trait value and the genetic value: $\mathbf{Y}_a = g_a + \varepsilon_a$ and $\mathbf{Y}_b = g_b + \varepsilon_b$. The genetic correlation is:

$$\rho_{a,b}^{Gen} = \frac{\sigma_{g_a,g_b}}{\sigma_{g_a}\sigma_{g_b}},$$

where σ_{g_a,g_b} is the genetic covariance between two traits and σ_{g_a}, σ_{g_b} are standard deviations of two traits in the population. $\rho_{a,b}^{Gen}$ ranges from –1 to 1, where 0 means genetic effects on one trait are independent of the other, and 1 indicates identical genetic influences on both traits [35].

Genetic correlation can be estimated in various ways, including the multi-trait model [34]. For two traits, the estimated genetic value's variance-covariance matrix $\hat{\mathbf{\Omega}}_{MT}$ is a 2x2 symmetric matrix. The estimated genetic correlation is derived by converting the genetic covariance matrix into a correlation matrix:

$$\hat{\rho}_{a,b}^{Gen} = \frac{\hat{\mathbf{\Omega}}_{MT_{1,2}}}{\sqrt{\hat{\mathbf{\Omega}}_{MT_{1,1}}\hat{\mathbf{\Omega}}_{MT_{2,2}}}}.$$

4

Bayesian method for MT ssGBLUP

GBLUP makes the Assumption 3.4 that the effects of the random effects are normally distributed. However we may want to make different assumptions about the distribution of random effects to include domain knowledge from stakeholders. Bayesian methods give us freedom to incorporate such prior assumptions into our analysis.

4.1. Prior Distributions for Random Effects

For now, we ignore fixed effects and focus on random effects: $\mathbf{Y}_i = \mu + \sum_{j=1}^{q} \mathbf{Z}_{ij} u_j + \varepsilon_i$,where $\mathbf{Z}_{ij}$ is the genotype of the i^{th} individual at the j^{th} marker and u_j is the corresponding marker effect. Recall that N is the number of individuals and q is the number of random effects. Let ω be the vector of hyperparameters. μ is commonly assigned a flat prior so $f(\mu) \propto 1$. We begin with the framework of standard Bayesian linear models in quantitative genetics. From

$$
\begin{aligned}
f(\mu, u, \sigma_\varepsilon^2, \mathbf{Y}, \omega) &= f(\mu, u, \sigma_\varepsilon^2 \mid \mathbf{Y}, \omega) f(\mathbf{Y}, \omega) \\
&= f(\mu, \mathbf{Z}, \sigma_\varepsilon^2, \mathbf{Y} \mid \omega) f(\omega) \\
&= f(\mathbf{Y} \mid \mu, u, \sigma_\varepsilon^2, \omega) f(\mu, u, \sigma_\varepsilon^2 \mid \omega) f(\omega),
\end{aligned}
$$

we get

$$\begin{aligned}
f(\mu, u, \sigma_\varepsilon^2, \mid \mathbf{Y}, \omega) &= f(\mathbf{Y} \mid \mu, u, \sigma_\varepsilon^2, \omega) f(\mu, u, \sigma_\varepsilon^2 \mid \omega) f(\omega) f^{-1}(\mathbf{Y}, \omega) \\
&\propto f(\mathbf{Y} \mid \mu, u, \sigma_\varepsilon^2) f(\mu, u, \sigma_\varepsilon^2 \mid \omega) \\
&= f(\mathbf{Y} \mid \mu, u, \sigma_\varepsilon^2) f(u \mid \mu, \sigma_\varepsilon^2, \omega) f(\mu, \sigma_\varepsilon^2 \mid \omega) \\
&= f(\mathbf{Y} \mid \mu, u, \sigma_\varepsilon^2) f(u \mid \mu, \sigma_\varepsilon^2, \omega) f(\mu) f(\sigma_\varepsilon^2) \\
&\propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{Y}_i \mid \mu + \sum_{j=1}^{q} \mathbf{Z}_{ij} u_j, \sigma_\varepsilon^2) \prod_{j=1}^{q} f(u_j \mid \omega) p(\sigma_\varepsilon^2).
\end{aligned}$$

We provide the meanings of three important parts mentioned above:

- $f(\mu, u, \sigma_\varepsilon^2 \mid \mathbf{Y}, \omega)$ is the posterior density of unknowns $\{ \mu, u, \sigma_\varepsilon^2 \}$ given phenotype and hyperparameters.
- $f(\mathbf{Y} \mid \mu, u, \sigma_\varepsilon^2) = \prod_{i=1}^{N} \mathcal{N}(\mathbf{Y}_i \mid \mu + \sum_{j=1}^{q} \mathbf{Z}_{ij} u_j, \sigma_\varepsilon^2)$ is the conditional density of phenotypic data given the unknowns.
- $f(\mu, u, \sigma_\varepsilon^2 \mid \omega) \propto \prod_{j=1}^{q} f(u_j \mid \omega) f(\sigma_\varepsilon^2)$ is the joint prior density of model unknowns.

The prior distribution of random effects $p(u_j \mid \omega)$ influences variable selection and shrinkage by determining its extent and type. Key features include mass near zero and tail thickness. Common informative priors, ordered by increasing mass peak at zero and tail weight, are Gaussian, heavy tail, Spike-Slab, and Point of Mass & Slab [15]. The uninformative prior is presented in this section, while informative priors are detailed in Appendix C.

Uninformative priors: A uninformative prior provides vague information about the variable, used when prior knowledge is lacking. It distributes probability widely, making it weakly informative, such as a normal prior with large variance [36].

4.2. Prior Distributions for Covariance Parameters

The error terms ε are assumed to be independent and identically distributed, each row following a multivariate normal distribution with zero mean and covariance matrix $\mathbf{R}$ (Assumption 3.8). Commonly used covariances structures are inverse Wishart, spherical and diagonal [37]. We cover the inverse Wishart distribution in this section. Details of spherical and diagonal priors are in Appendix C.

Inverse Wishart Distribution: For inverse Wishart distribution, we follow the work given by Zhang [38]. The inverse Wishart distribution is a popular unstructured prior because it is conjugate to normal data and ensures positive definite covariance matrix. The density function of an inverse Wishart distribution $IW(\mathbf{V}, m)$ with the scale matrix $\mathbf{V}$ and the degrees of freedom m for a $p \times p$ variance-covariance matrix $\mathbf{\Sigma}$ is

$$p(\mathbf{\Sigma}) = IW(\mathbf{V}, m) = \frac{|\mathbf{V}|^{m/2} |\mathbf{\Sigma}|^{-(m+p+1)/2} \exp\left[-\operatorname{tr}\left(\mathbf{V}\mathbf{\Sigma}^{-1}\right)/2\right]}{2^{mp/2}\Gamma(m/2)}. \tag{4.1}$$

The inverse Wishart distribution is a generalization of the inverse gamma distribution to multiple dimensions. If $\mathbf{V} = 2\alpha$, $m = 2\beta$ and $p = 1$, Equation (4.1) changes to the inverse gamma distribution:

$$f(\mathbf{\Sigma}; \alpha, \beta) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} \mathbf{\Sigma}^{-\alpha-1} exp(-\beta/\mathbf{\Sigma}),$$

where α is the shape parameter and β is the scale parameter. The mean of inverse Wishart distribution is

$$E(\mathbf{\Sigma}) = \frac{\mathbf{V}}{m - p - 1}$$

and the variance of each element of $\mathbf{\Sigma} = \left(\sigma_{ij}\right)$ is

$$\operatorname{Var}\left(\sigma_{ij}\right) = \frac{(m-p+1)v_{ij}^2 + (m-p-1)v_{ii}v_{jj}}{(m-p)(m-p-1)^2(m-p-3)}.$$

Especially,

$$\operatorname{Var}\left(\sigma_{ii}\right) = \frac{2v_{ii}^2}{(m-p-1)^2(m-p-3)}.$$

Theorem 3. *The inverse Wishart distribution is conjugate to normal distribution.*

Proof. Let $\mathbf{X} := (\mathbf{x}_1, \ldots, \mathbf{x}_t)$ denote a vector of t variables following the multivariate normal distribution:

$$\mathbf{X} \mid \mathbf{\Sigma} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Sigma}),$$

where the mean vector $\boldsymbol{\mu} = \mathbf{0}$ and the variance-covariance matrix $\mathbf{\Sigma}$. The density function is

$$f(\mathbf{X} \mid \mathbf{\Sigma}) = (2\pi)^{-t/2} |\mathbf{\Sigma}|^{-1/2} \exp\left(-\frac{1}{2}\mathbf{X}^T \mathbf{\Sigma}^{-1} \mathbf{X}\right).$$

The likelihood function for $\mathbf{\Sigma}$ is

$$\begin{aligned} L(\mathbf{\Sigma} \mid \mathbf{X}) = p(\mathbf{X} \mid \mathbf{\Sigma}) &\propto |\mathbf{\Sigma}|^{-t/2} \exp\left(-\frac{1}{2}\sum_{i=1}^{t} \mathbf{x}_i^T \mathbf{\Sigma}^{-1} \mathbf{x}_i\right) \\ &= |\mathbf{\Sigma}|^{-t/2} \exp\left[-\frac{1}{2} \operatorname{tr}\left(\sum_{i=1}^{t} \mathbf{x}_i \mathbf{x}_i^T \mathbf{\Sigma}^{-1}\right)\right] \\ &= |\mathbf{\Sigma}|^{-t/2} \exp\left[-\frac{t}{2} \operatorname{tr}\left(\mathbf{S}\mathbf{\Sigma}^{-1}\right)\right], \end{aligned}$$

where $\mathbf{S} = \sum_{i=1}^{t} \mathbf{x}_i \mathbf{x}_i^T / t$ is the biased sample covariance matrix. When the prior distribution is inverse

Wishart $IW(\mathbf{V}_0, m_0)$, the posterior distribution of $\mathbf{\Sigma}$ is

$$\begin{aligned} p(\mathbf{\Sigma} \mid \mathbf{X}) &\propto p(\mathbf{X} \mid \mathbf{\Sigma}) p(\mathbf{\Sigma}) \\ &\propto |\mathbf{\Sigma}|^{-t/2} \exp\left[-\frac{t}{2} \operatorname{tr}\left(\mathbf{S}\mathbf{\Sigma}^{-1}\right)\right] |\mathbf{\Sigma}|^{-\left(m_0+p+1\right)/2} \exp\left[-\operatorname{tr}\left(\mathbf{V}_0\mathbf{\Sigma}^{-1}\right)/2\right] \\ &= |\mathbf{\Sigma}|^{-\left(t+m_0+p+1\right)/2} \exp\left\{-\frac{1}{2} \operatorname{tr}\left[\left(t\mathbf{S}+\mathbf{V}_0\right)\mathbf{\Sigma}^{-1}\right]\right\}. \end{aligned}$$

Therefore, the posterior distribution for $\mathbf{\Sigma}$ is also an inverse Wishart distribution: $\mathbf{\Sigma} \mid \mathbf{X} \sim IW\left(t\mathbf{S} + \mathbf{V}_0, t + m_0\right)$.

□

The posterior mean of $\mathbf{\Sigma}$ is

$$\begin{aligned} E(\mathbf{\Sigma} \mid \mathbf{X}) &= \frac{t\mathbf{S} + \mathbf{V}_0}{t + m_0 - p - 1} \\ &= \frac{t}{t + m_0 - p - 1}\mathbf{S} + \left(1 - \frac{t}{t + m_0 - p - 1}\right)\frac{\mathbf{V}_0}{m_0 - p - 1}. \end{aligned}$$

A weighted average of the sample covariance matrix $\mathbf{S}$ and the prior mean $\mathbf{V}_0/\left(m_0 - p - 1\right)$ is the posterior mean. The posterior mean will approach to the sample mean given fixed m_0 and p as the sample size t increases [38].

4.3. Gibbs Sampler

In practice, it is always difficult to integrate out other parameters to calculate the posterior distribution. There are a number of ways to overcome this problem:

- A conjugate prior provides a closed-form posterior expression. It results in a recognized posterior distribution when combine with a particular distribution for the data [39]. For instance, the inverse Wishart distribution is conjugate to the normal distribution.
- Numerical integration: Simpson's rule can be used if we calculate the height of the posterior distribution at every point [40].
- Variational Bayes: We can approximate the functions used to calculate the posterior with simpler functions and show that the resulting approximate posterior is close to true posterior [41].
- Simulation: If we can draw samples from the posterior distribution, samples can be used to approximate the distribution, for example, Markov Chain Monte Carlo (MCMC) [39].

We introduce MCMC as a method to sample from complex distributions. In a Markov chain, each state depends only on the current state. To obtain samples from the complex distribution, the chain's stationary distribution must match the target distribution. The Monte Carlo method is used when direct sampling from the distribution is not possible.

Consider a D dimensional posterior with parameters $\theta = (\theta_1, \theta_2, \dots, \theta_D)$. Denote the i^{th} sample by $\theta^i = (\theta_1^i, \theta_1^2 \dots, \theta_D^i)$. The chain starts from initial point θ^0. The sampling process goes according to transition probability of Markov chain $T(\theta^i \mid \theta^{i-1})$ which is the probability that the sample state θ^{i-1} switches to θ^i. Samples in the burn-in phase are discarded until they reach the stationary distribution, where the sample distribution approximates the target distribution. After the burn-in phase, Markov chain samples match the target distribution [42, 43].

Gibbs sampler is particularly well-adapted to sampling the posterior distribution, since posterior distributions are typically specified as a collection of conditional distributions. The basic idea of Gibbs sampler is to iterately sample from the conditional distribution $f(\theta_j \mid X, \theta_{-j})$ where θ_{-j} is θ without the j^{th} parameter. A new sampler from the joint posterior density is obtained by sampling from fully conditional density. Therefore, it is convenient when the fully conditional densities have closed form and are able to sample from. Besides, Gibbs sampler is a special case of acceptance-rejection method. In Gibbs sampler the new sample is always accepted with probability one. So it more efficient than regular acceptance-rejection method for example Metropolis-Hastings [44, 45].

Gibbs sampler has several advantages: there is no need to tune targeted distribution. New samples are always accepted which make it more efficient than acceptance-rejection method. However Gibbs sampler may be very slow if parameters are correlated because samples can only change in one dimension at each step [46].

Algorithm 1 Gibbs sampler

```
Input: $\theta^{(0)}$, initial values
Input: $N_{iterations}$, the number of iterations
Input: $N_{burnin}$, the number of iterations in burn-in period
for $i = 1$ to $N_{iterations}$ do
    for $j = 1$ to $D$ do
        Sample the components of $\theta^{(i+1)}$ in order, starting from the first component.
        Update $\theta_j^{(i+1)}$ according to the distribution specified by
            $$f(\theta_j^{(i+1)} | \theta_1^{(i+1)}, \dots, \theta_{j-1}^{(i+1)}, \theta_{j+1}^{(i)}, \dots, \theta_D^{(i)}).$$
    end for
end for
List = [ ]
for $i = N_{burnin} + 1$ to $N_{iterations}$ do
    List = List.append ($\theta^{(i)}$).
end for
Return: List
```

5

Statistical Learning Model

This chapter will cover three models: LASSO regression, random forest, and XGBoost, used for predicting conventional traits from digital traits. LASSO regression utilizes L^1 regularization for variable selection, while Random Forest and XGBoost are tree-based methods.

5.1. LASSO Regression

The least absolute shrinkage and selection operator (LASSO) was introduced in 1996 [47]. It assumes that the coefficients of the linear model are sparse which means only few of them are non-zero. Consider a regression problem with p independent variables and a single response $Y = X\beta$. The ordinary least squares estimator is

$$\hat{\beta}^{OLS} = \arg\min_{\beta} (Y - X\beta)^T (Y - X\beta).$$

The LASSO estimator is defined by

$$\hat{\beta}^{LASSO} = \arg\min_{\beta} \left[(Y - X\beta)^T (Y - X\beta) + \lambda \sum_{j=1}^{p} |\beta_j| \right],$$

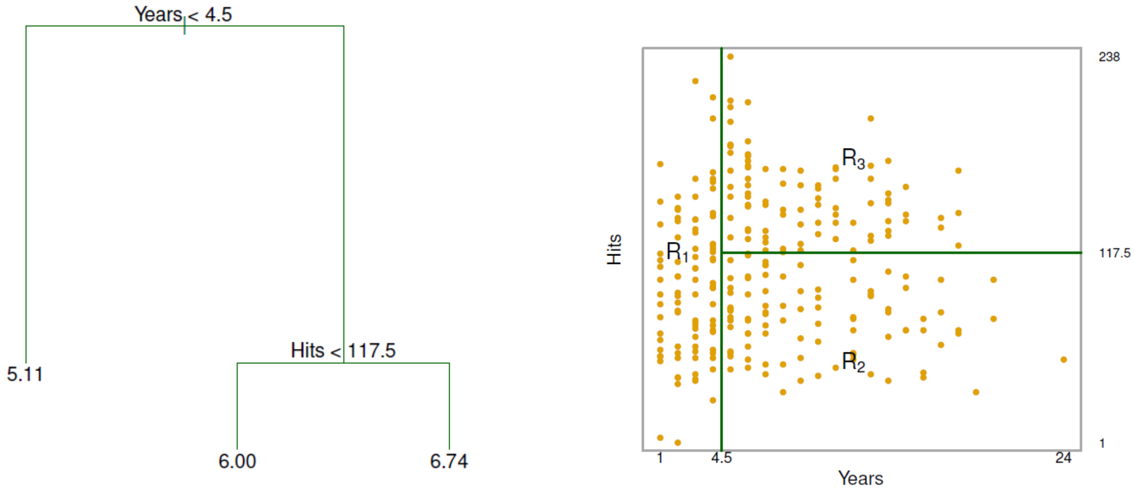
where λ is the regularization parameter. Lasso achieves variable selection by forcing the sum of the absolute value of the regression coefficients to be less than a fixed value, which forces certain coefficients to zero. Compared with ordinary least squares, the objective function of LASSO has an penalty term $\lambda \sum_{j=1}^{p} |\beta_j|$. λ is a hyperparameter that balances the tradeoff between bias and variance in the estimated coefficients $\hat{\beta}^{LASSO}$. As λ increases, the bias increases, and the variance decreases, leading to a simpler model with fewer parameters. $\hat{\beta}^{LASSO}$ is the same as $\hat{\beta}^{OLS}$ if $\lambda = 0$ [48, 49].

5.2. Decision Tree

We start with fundamental decision trees before introducing random forest and XGBoost. Decision trees can be applied to both regression and classification problems so they are also called classification and regression tree (CART) [49]. We will only focus on regression problems in this chapter.

We present the mechanism of regression trees with an example of predicting a baseball player's *Salary* using *Year* (years played in major leagues) and *Hits* (hits made in the previous year), utilizing the data available in the R dataset *Hitters* [48]. Figure 5.1a displays a regression tree. It splits data into three groups using a series of rules: the first is *Year*< 4.5, then players with $Year \geq 4.5$ are split by the second rule, *Hits* < 117.5.

Figure 5.1b illustrates the tree by showing the tree's three-region partition: $R_1 = \{X \mid Year < 4.5\}$, $R_2 = \{X \mid Year \geq 4.5, Hits < 117.5\}$, $R_3 = \{X \mid Year \geq 4.5, Hits \geq 117.5\}$. The tree predicts new players' *Salary* by the mean *Salary* of players in the same region.



(a) Regression tree

(b) Partition of the regression tree

Figure 5.1: Prediction of *Salary* by *Year* and *Hits* [48]

The dataset contains p inputs and a response across N observations: (x_i, y_i) where $i = 1, 2, \ldots, N$ with $x_i = (x_i^1, x_i^2, \ldots, x_i^p)$. A decision tree splits the predictor space into $M \in \mathbb{Z}^+$ non-overlapped regions. We model the response by

$$f(x) = \sum_{m=1}^{M} ave(y_i \mid x_i \in R_m)\mathbf{I}(x \in R_m), \tag{5.1}$$

where $ave(y_i \mid x_i \in R_m)$ is the average y of the points with in the region R_m. The goal is to find regions $\{R_1 \ldots R_M\}$ to minimize the loss function:

$$\sum_{i=1}^{N} (y_i - f(x_i))^2. \tag{5.2}$$

The regions can be in any shape but we choose to use the high-dimensional rectangles for simplicity. It is a waste of computing resources to enumerate all the possible tree structures. A top-down, greedy algorithm is used to find the best splitting point. It is top-down because it starts from the top of the tree and then splits the predictor space successively. It is greedy because at each split, the best split is decided at that particular step without looking forwards. The top-down and greedy algorithm can build a tree in a relatively short time at the cost of missing the best tree structure [50].

To split a predictor space we need a predictor and a cutpoint (X_j, s) which leads to the largest reduction of Equation (5.2). Assume that $R_1(j, s)$ and $R_2(j, s)$ are the instance sets after splitting R_0, where $R_1(j, s) = \{X \mid X_j < s\}$, $R_2(j, s) = \{X \mid X_j \geq s\}$ and $R_0 = R_1(j, s) \cup R_2(j, s)$. j and s are given by the following equation:

$$\min_{j,s}[\sum_{x_i \in R_1(j,s)} (y_i - ave(y_i \mid x_i \in R_1(j, s)))^2 + \sum_{x_i \in R_2(j,s)} (y_i - ave(y_i \mid x_i \in R_2(j, s)))^2]. \tag{5.3}$$

After splitting the root into two regions, we split one of the regions next to minimize (5.2). This results in three regions, and we search for the best splits within them. The process continues until a predefined stopping criterion, such as maximum tree depth or minimum leaf samples, is met. Once regions $\{R_1, \ldots, R_J\}$ are established, predictions are made using the mean of ys in the corresponding region [50].

It is important to determine the final tree size in decision tree modeling. A large tree risks overfitting, while a small tree may miss important patterns. Post-pruning is to grow the tree as big as possible then remove nodes that do not provide enough information. This process aims to decrease the size of the tree while keeping predictive accuracy [48, 50].

5.3. Bagging and Random Forest

Bagging aggregates multiple independent base models to improve stability and accuracy of machine learning algorithms. It reduces variance and avoids overfitting by averaging multiple predictions for the final result [51]. It can be used with any type of base models, we will focus on bagging of CARTs.

The random forest is a bagging and tree-based model devised in 2000 with many advantages compared with single CART: less variance, less overfitting, and better overall performance [51]. Figure 5.2 shows the flowchart of the random forest algorithm. Multiple CARTs grow from different bootstrap samples. Each of them makes predictions independently then predictions are aggregated as final output.

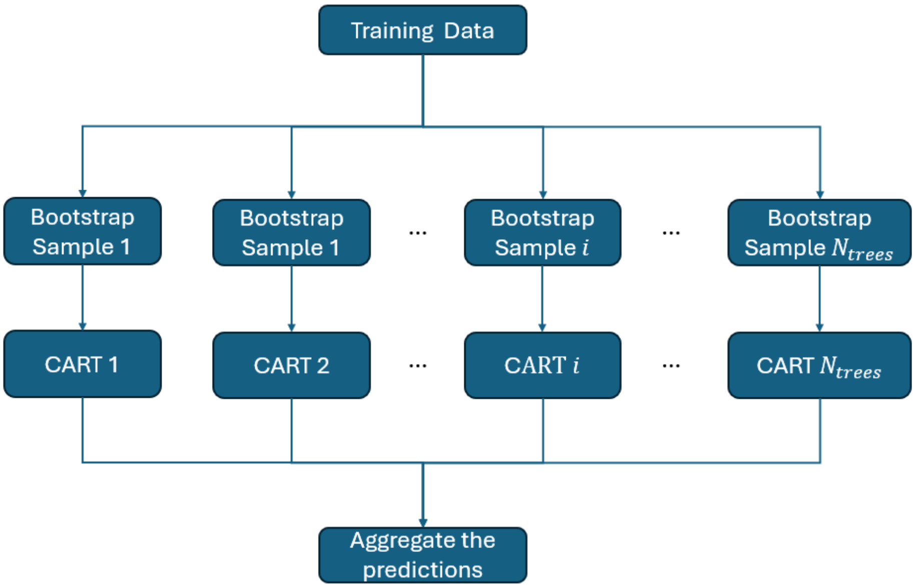


Figure 5.2: Flowchart of the random forest

Define the correlation between two trees f_i and f_j as $\rho = \frac{\text{Cov}(f_i(x), f_j(x))}{\sigma_{f_i(x)}\sigma_{f_j(x)}}$. The variance of $f(x)$ is

$$\begin{aligned}
\text{Var}(f(x)) &= \text{Var}\left(\frac{1}{N_{trees}}\sum_{j=1}^{N_{trees}} f_j(x)\right) \\
&= \frac{1}{N_{trees}{}^2}\sum_{i=1}^{N_{trees}}\sum_{j=1}^{N_{trees}} \text{Cov}\left(f_i(x), f_j(x)\right) \\
&= \frac{1}{N_{trees}{}^2}\sum_{i=1}^{N_{trees}}\left(\sum_{j\neq i}^{N_{trees}} \text{Cov}\left(f_i(x), f_j(x)\right) + \text{Var}\left(f_i(x)\right)\right) \\
&= \frac{1}{N_{trees}{}^2}\sum_{i=1}^{N_{trees}} \left((N_{trees}-1)\sigma^2\rho + \sigma^2\right) \\
&= \frac{N_{trees}(N_{trees}-1)\rho\sigma^2 + N_{trees}\sigma^2}{N_{trees}{}^2} \\
&= (\rho + \frac{1-\rho}{N_{trees}})\sigma^2. \qquad (5.4)
\end{aligned}$$

Equation (5.4) shows that Var($f(x)$) decreases as ρ decreases. The random forest uses two methods to reduce the correlation between trees by adding randomness to the training set: bootstrap and feature selection. The samples in training sets are different between trees because of bootstrap. Besides, each bootstrap sample only include a subset of features in training data.

After growing all trees, the model can make predictions on new observations. When there is one new observation, the output of random forest regression is the average prediction from all trees [50, 52]. The pseudo code of the random forest is displayed in Algorithm 2. The notations used in Algorithm 2 and 3 are as following:

- $\mathcal{D} := (\mathbf{X}, \mathbf{Y}) = (x_i, y_i)$ where $i \in \{1, 2, \ldots N_{train}\}$ is training data;
- $\mathcal{D}_{bootstrap} := (\mathbf{X}_{bootstrap}, \mathbf{Y}_{bootstrap})$ is the bootstrap data used to grow a particular CART;
- N_{trees} is the number of CARTs;
- $N_{subfeatures}$ is the number of features in $\mathbf{X}_{bootstrap}$.

Algorithm 2 Random Forest

```
Input: $\mathcal{D}$, $N_{trees}$, $N_{subfeatures}$, the stopping criterion and a new observation $x$
for $j = 1$ to $N_{trees}$ do
    Generate a bootstrap sample $\mathcal{D}_{bootstrap}$ with $N_{subfeatures}$ features.
    Start with a single instance set containing all data points from $\mathcal{D}_{bootstrap}$.
    while the stopping criterion is not satisfied do
        For each feature, calculate the gain from splitting at each potential point by Equation (5.3).
        Choose the feature and split point that give the maximum gain.
        Split the instance set into two subsets.
    end while
    Compute the predicted value at $x$ from the $j^{th}$ tree $f^j(x)$ by Equation (5.1).
end for
Compute the predicted value at $x$ from the random forest by $f(x) = \frac{1}{N_{trees}} \sum_{j=1}^{N_{trees}} f^j(x)$.
Return: $f(x)$
```

5.3.1. Tree's Feature Importance from Mean Decrease in Impurity

Split a regression tree T at instance set t. A proposed split s for variable X_j divides t into left and right subsets t_L and t_R based on $X_j \leq s$ or $X_j > s$. The sample variance within each instance set determines its impurity. The impurity of t is defined as

$$\widehat{\Delta}(t) = \frac{1}{N} \sum_{X_i \in t} \left(Y_i - \bar{Y}_t\right)^2$$

where $\bar{Y}_t$ is the sample mean for t and N is the sample size of t. The within sample variance for subsets are

$$\widehat{\Delta}(t_L) = \frac{1}{N_L} \sum_{i \in t_L} \left(Y_i - \bar{Y}_{t_L}\right)^2 \text{ and } \widehat{\Delta}(t_R) = \frac{1}{N_R} \sum_{i \in t_R} \left(Y_i - \bar{Y}_{t_R}\right)^2,$$

where $\bar{Y}_{t_L}$ is the sample mean for t_L and N_L is the sample size of t_L (similar definitions apply to t_R). The decrease in impurity because of the split s for X_j is

$$\widehat{\Delta}(s, t) = \widehat{\Delta}(t) - \left[\widehat{p}(t_L) \widehat{\Delta}(t_L) + \widehat{p}(t_R) \widehat{\Delta}(t_R)\right],$$

where $\hat{p}(t_L) = N_L/N$ and $\hat{p}(t_R) = N_R/N$ are the proportions of observations in t_L and t_R, respectively.

For each feature, sum the impurity reductions at all nodes where it was split across all forest trees.

Normalize each feature's importance score by dividing it by the total importance score. It gives a percentage that represents the importance of each feature relative to the others [53].

5.4. Boosting and XGBoost

Similar to tree bagging, tree boosting also uses a series of decision trees: $f^1, f^2, \dots f^{N_{trees}}$. Recall that in bagging, each tree grows on a bootstrap sample and independent of the other trees. In boosting, each tree is grown to correct the prediction error from previously trees so the trees are sequential and dependent [48]. Figure 5.3 displays the flowchart of boosting. The output of a boosting with N_{trees} CARTs is the sum of predicted values from all CARTs: $f(x) = \sum_{j=1}^{N_{trees}} f^j(x)$.

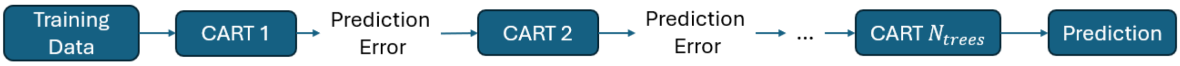


Figure 5.3: Flowchart of decision tree boosting

Figure 5.4 is an example to show how a XGBoost makes prediction [54]. The dependent variable is the attitude towards video games, and the independent variables are age, gender, and whether or not to use a computer daily. The first tree uses age and gender as independent variables, and the second tree uses whether or not to use a computer every day. The prediction is the sum of the predictions of the each tree. The boy gets 2 and 0.9 from the first and second tree respectively so predicted value of the boy is 2.9.

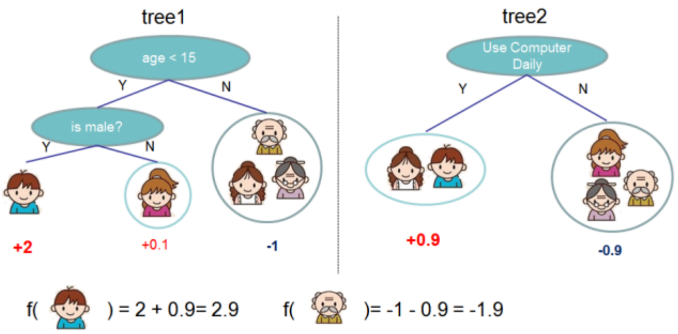


Figure 5.4: Example of decision tree boosting [54]

Algorithm 3 Boosting

```
Input: $\mathcal{D}$, $N_{trees}$, the stopping criterion and a new observation $x$
Set $f(x) = 0$ and $\varepsilon_i = y_i$ where $i \in \{1, 2, \dots, N_{train}\}$.
for $j = 1$ to $N_{trees}$ do
    Grow a tree $f^j$ on the training data $(\mathbf{X}, \varepsilon)$ and compute the predicted value from the $j^{th}$ tree $f^j(x)$ like lines 5-10 in Algorithm 2.
    Update $f(x)$ by $f(x) \leftarrow f(x) + f^j(x)$.
    Update $\varepsilon_i$ by $\varepsilon_i \leftarrow \varepsilon_i - f^j(x_i)$ for all $i$.
end for
Return: $f(x)$
```

XGBoost, proposed in 2006 as eXtreme Gradient Boosting [54], follows the Algorithm 3 at basic level and improves accuracy and speed through specific methods. A series of methods are used to avoid overfitting: regularization, shrinkage and feature subsampling. Regularization in XGBoost penalizes the number of leaves T and the leaf score vector w through two main types: L^1 and L^2. L^1 regularization reduces feature coefficients to zero for variable selection, while L^2 regularization shrinks coefficients to handle multicollinearity [49].

We denote loss function as l which measures the difference between the predicted value $\hat{y}_i$ and the real value y_i. The objective function of XGBoost is

$$\sum_{i=1}^{N_{train}} l\left(\hat{y}_i, y_i\right) + \sum_{j=1}^{N_{trees}} \Omega\left(f_j\right), \text{ where } \Omega(f) = \gamma T + \frac{1}{2}\lambda\|w\|^l \text{ and } l \in \{1, 2\}. \tag{5.5}$$

Equation (5.5) consists of two parts: the difference between predicted and real values in training data and penalty for model complexity. The model will only focusing on achieving minimum loss functions on training set if regularization is not used.

Shrinkage adjusts new tree weights by a factor λ each step, updating line 5 in Algorithm 3 to: Update $f(x)$ by $f(x) \leftarrow f(x) + \lambda f^j(x)$. This reduces each tree's impact, making boosting more conservative. Feature subsampling, used in XGBoost and also in random forests, generates different training sets for different trees.

Compared with the bagging, one of the disadvantages of the boosting is the speed. This is because base models need to be built one by one in boosting while they can be built at the same time in bagging [49]. To speed up the training process, the XGBoost implements an approximate algorithm which aims to speed up the tree-building process. It finds the best splitting point by only checking quantiles of the feature. Figure 5.5 is an example where we have 40 data points. The model will compare 39 different splitting points if it check them one by one,. However the model only compares 9 different points if it only pays attention to the feature values at 10%, 20%, . . . , 90% quantiles. The one with largest loss reduction is the optimal splitting point for this feature. The approximate algorithm is in 7-12 lines of Algorithm 4.

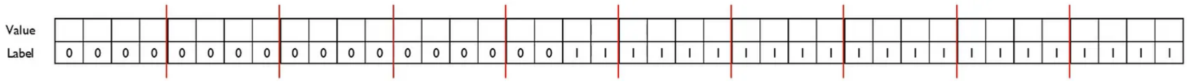


Figure 5.5: Approximate Algorithm

Algorithm 4 Boosting plus Approximate Algorithm

```
Input: $\mathcal{D}$, $N_{trees}$,$N_{subfeatures}$ ,the stopping criterion and a new observation $x$
Set $f(x) = 0$ and $\varepsilon_i = y_i$ where $i \in \{1, 2, \ldots, N_{train}\}$.
for $j = 1$ to $N_{trees}$ do
    Generate a bootstrap sample $\mathcal{D}_{bootstrap}$ with $N_{subfeatures}$ features.
    Start with a single instance set containing all data points from $\mathcal{D}_{bootstrap}$.
    while the stopping criterion is not satisfied do
        for $j = 1$ to $N_{features}$ do
            for $s$ = values of $j^{th}$ features at 10%, 20%, . . . , 90% quantiles do
                Calculate split value of (j, s) by
                $$\sum_{x_i \in R_1(j,s)} (\varepsilon_i - ave(\varepsilon_i \mid x_i \in R_1(j,s)))^2 + \sum_{x_i \in R_2(j,s)} (\varepsilon_i - ave(\varepsilon_i \mid x_i \in R_2(j,s)))^2.$$
            end for
        end for
        Choose the feature and split point that give the minimum split value.
        Split the instance set into two subsets.
    end while
    Compute the predicted value from the $j^{th}$ tree $f^j(x)$ by Equation (5.1).
    Update $f(x)$ by $f(x) \leftarrow f(x) + \lambda f^j(x)$.
    Update $\varepsilon_i$ by $\varepsilon_i \leftarrow \varepsilon_i - \lambda f^j(x_i)$ for all $i$.
end for
Return: $f(x)$
```

6

Blocky Pepper Data

This research studied 315 blocky pepper cultivars across 348 plots, all planted in the same field and season, with some cultivars appearing in more than one plot. Blocky peppers were red, orange, or yellow. Each plot's data included plot name, cultivar, trait names, and values. Traits were either conventional (subjective ratings) or digital (continuously recorded by ScaleCam). Table 6.1 lists 21 digital traits involving color, size, and shape, identified by "D" in their names. Table 6.2 includes seven conventional traits covering the same characteristics plus other quality features, indicated by "C." All cultivars were assessed for both types of traits.

6.1. Background

We grew blocky peppers in a field.

- The field was divided into N_{plot} plots with plot index i and $i \in \{1, 2, \ldots, N_{plot}\}$.
- Each plot was harvested $N_{harvest}$ times with harvest index h where $h \in \{1, 2, \ldots, N_{harvest}\}$.
- In plot i, we had several plants. blocky peppers were collected in the plot and measured.
- The number of fruits in the plot i^{th} and harvest h^{th} are indicated by $N_{fruit}(i, h)$. The index of fruits is p, $p \in \{1, 2, \ldots, N_{fruit}(i, h)\}$.
- Each fruit was measured for N_{trait} traits. The index of traits is t, $t \in \{1, 2, \ldots, N_{trait}\}$.
- There were $N_{cultivar}$ cultivars in this field. Fruits in a plot were from the same cultivar. A cultivar appeared in $N_{plot}(c)$ plots where c was the index of cultivar $c \in \{1, 2, \ldots, N_{cultivar}\}$.
- Cultivar: $\{1, 2, \ldots, N_{plot}\} \longrightarrow \{1, 2, \ldots, N_{cultivar}\}$.

Trait Group	Description	Trait Names
Color	Color bands of RGB color model from RGB camera [1]	ColorD1
		ColorD2
		ColorD3
		ColorD4
		ColorD5
Size	Traits about ellipsoid, triangle, shape index, rectangular, etc	SizeD1
		SizeD2
		SizeD3
		SizeD4
		SizeD5
		SizeD6
		SizeD7
		SizeD8
Shape	Traits about area, perimeter, height, width, etc	ShapeD1
		ShapeD2
		ShapeD3
		ShapeD4
		ShapeD5
		ShapeD6
		ShapeD7
		ShapeD8

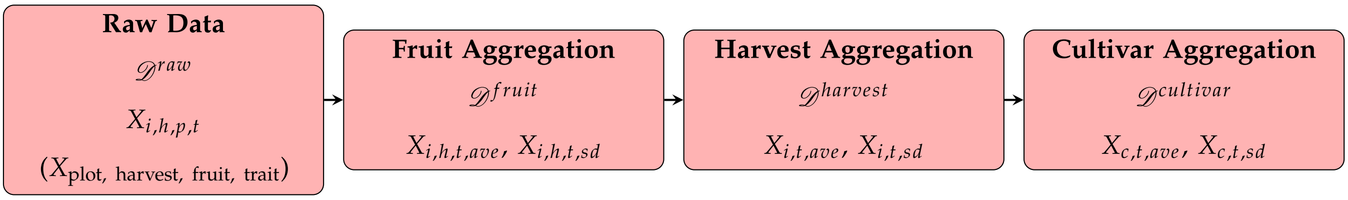
Table 6.1: Digital trait list

Trait Group	Description	Trait Names	Range
Color	Exterior Color Rating	ColorC	$\{k \in \mathbb{N} : 1 \leq k \leq 9\}$
Shape	Shape Rating	ShapeC1	
	Shape Uniformity Rating	ShapeC2	
Size	Size Rating	SizeC1	
	Size Uniformity Rating	SizeC2	
Other Quality	Firmness Rating	OtherC1	
	Cracking Rating	OtherC2	

Table 6.2: Conventional trait list

[1] The RGB color model is an additive color model in which the red, green and blue primary colors of light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue [55].

6.2. Aggregation



The original data $\mathscr{D}^{raw}$ contained trait values for each fruit $X_{i,h,p,t}$. $X_{i,h,p,t}$ indicated the trait value of the t^{th} trait of the p^{th} fruit from the h^{th} harvest of the i^{th} plot was X. $\mathscr{D}^{fruit}$ was calculated by aggregating the trait values of different fruits in the same plot and the same harvest in $\mathscr{D}^{fruit}$.

$$X_{i,h,t,ave} = \frac{1}{N_{fruit}(i,h)} \sum_{p=1}^{N_{fruit}(i,h)} X_{i,h,p,t},$$

$$X_{i,h,t,sd} = SD(\{X_{i,h,p,t}\}) \text{ where } p \in \{1, 2, \ldots, N_{fruit}(i,h)\}.$$

$\mathscr{D}^{harvest}$ was calculated by aggregating trait values of different harvests from the same plot in $\mathscr{D}^{fruit}$:

$$X_{i,t,ave} = \frac{1}{N_{harvest}(i)} \sum_{h=1}^{N_{harvest}(i)} X_{i,h,t,ave},$$

$$X_{i,t,sd} = \frac{1}{N_{harvest}(i)} \sum_{h=1}^{N_{harvest}(i)} X_{i,h,t,sd}.$$

$\mathscr{D}^{cultivar}$ was calculated by aggregating the trait values of plots with the same cultivar in $\mathscr{D}^{harvest}$:

$$X_{c,t,ave} = \frac{1}{N_{plot}(c)} \sum_{i=1}^{N_{plot}(c)} X_{i,h,t,ave},$$

$$X_{c,t,sd} = \frac{1}{N_{plot}(c)} \sum_{i=1}^{N_{plot}(c)} X_{i,h,t,sd}.$$

Only $\mathscr{D}^{cultivar}$ will be used in following analysis and modeling.

6.3. Descriptive Analysis

Figure 6.1 displays the estimated probability density functions of conventional traits using Gaussian kernel density. The OtherC2 trait, with values mostly around one, was abandoned in further analysis. This step was 'Distribution Check' in the workflow (Figure 1.1). The estimated probability density

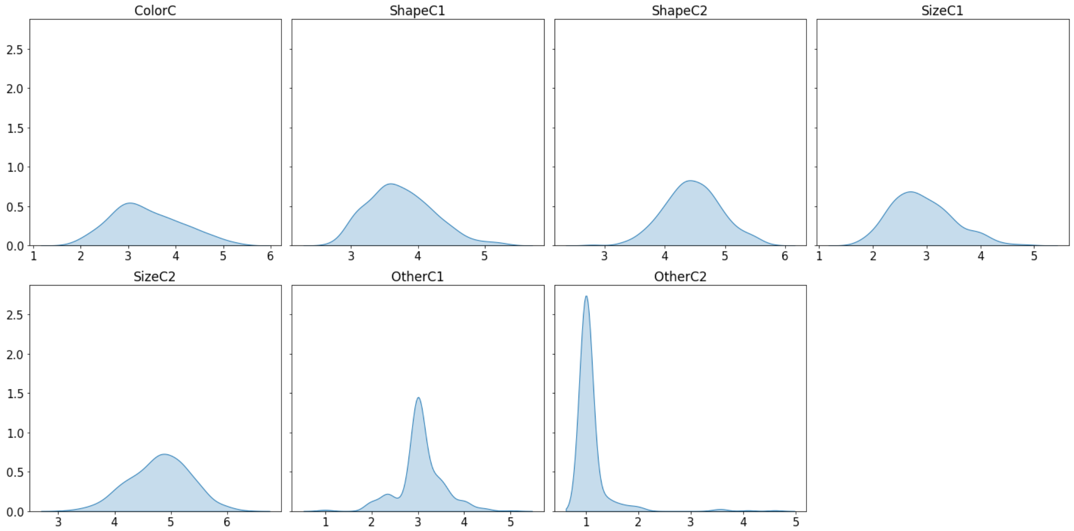


Figure 6.1: Estimated probability density functions of conventional trait values

functions of digital traits and a statistical summary table are available in Appendix A.

Next, we examine Figure 6.2, a hierarchically-clustered heatmap of Pearson correlations among traits, preceded by an overview of the clustering method. Euclidean distance is used to calculate the distance between two points. Let p have coordinates (p_1, p_2) and q have coordinates (q_1, q_2). The Euclidean distance between p and q is

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}.$$

The hierarchical clustering method uses the nearest point algorithm to calculate the distance $d(s, t)$ between clusters s and t: $d(s, t) = \min_{i,j}(dist(u[i], v[j]))$ for each point i in cluster u and each point j in cluster v. The cluster process starts with a group of points and each point is one cluster. When two clusters s and t are combined into a single cluster u, s and t are eliminated and u is introduced into the group. The algorithm stops when only a single cluster remains and this cluster becomes the root.

In the dendrogram, conventional traits except *ShapeC1* were broadly clustered, indicating larger Euclidean distances between conventional and digital traits than within conventional traits. Notably, conventional shape trait *ShapeC1* clustered with digital shape traits. Digital traits from the same groups (Color, Shape, Size) tended to cluster together.

The heatmap of Pearson correlations showed positive correlations among the standard deviations for digital color, size, and shape, indicating uniformity. Unexpectedly, size (*SizeD7*, *SizeD3*, *SizeD8*) and shape (*ShapeD2*, *ShapeD4*, *ShapeD1*) were also correlated, revealing a new finding. The heatmap and dendrogram always agreed with each other: closely grouped traits were more correlated.

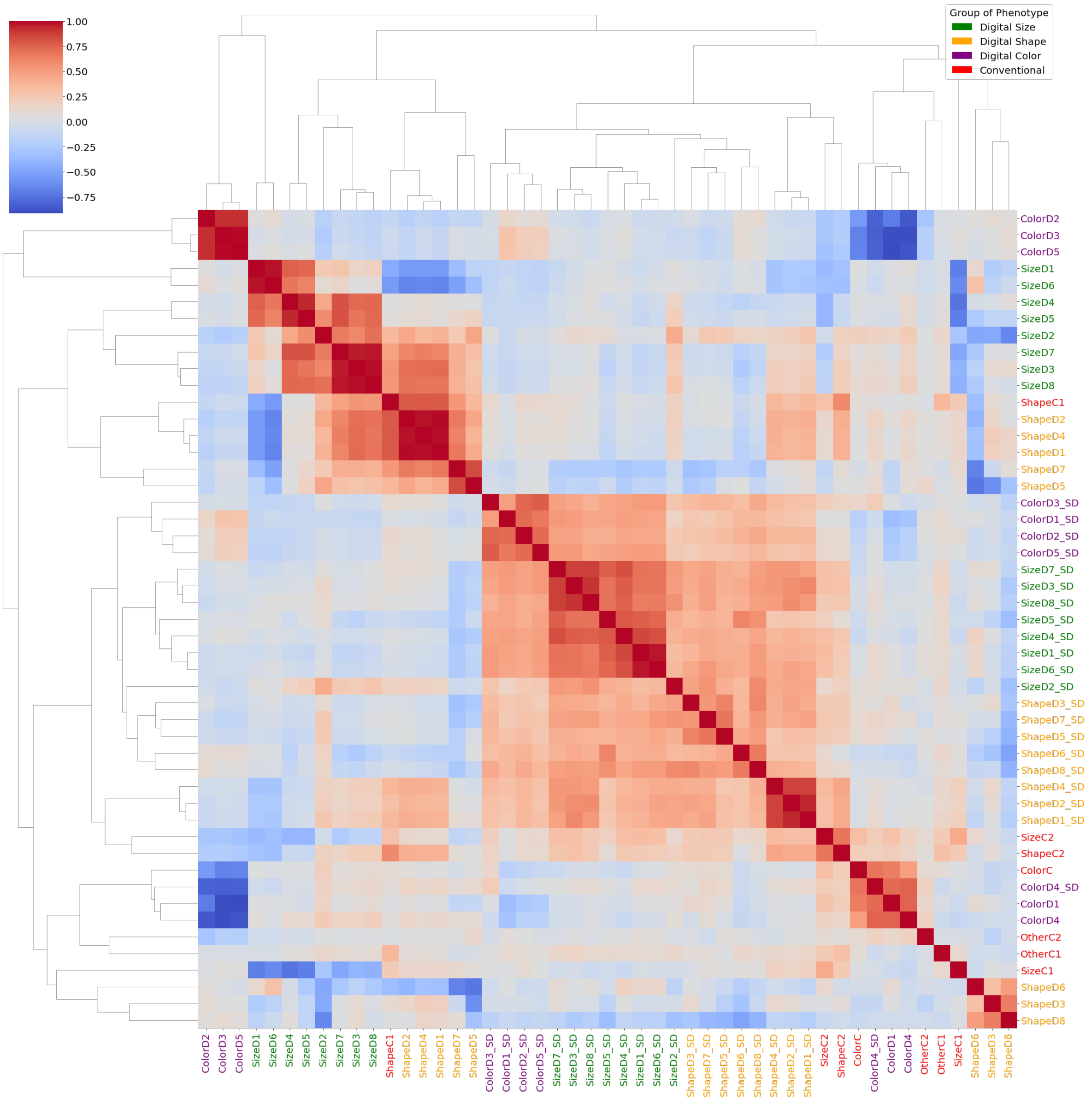


Figure 6.2: Hierarchically-clustered heatmap of Pearson correlations among traits

7

Calculation of the Estimated Genetic Correlation $\hat{\rho}^{Gen}$

7.1. Use of Multi-Trait Model and Gibbs Sampler

We used the multi-trait model to estimate genetic correlations $\hat{\rho}^{Gen}$ via the BGLR package function *multitrait()* [37]. Flat prior and inverse Wishart prior were selected for random effects and covariance structure because stakeholders did not have enough prior knowledge. The genetic covariance matrix $\widehat{\mathbf{\Omega_{MT}}}$ was derived from the posterior mean using Gibbs sampler.

Conventional and digital trait sets are represented as $\mathbf{T}_{con}$ and $\mathbf{T}_{digital}$. A conventional and a digital trait are represented as $\mathbf{t}_{con} \in \mathbf{T}_{con}$ and $\mathbf{t}_{digital} \in \mathbf{T}_{digital}$. The bi-trait model in Equation (7.1), a special case of the multi-trait model, was used to estimate the genetic correlation between $\mathbf{t}_{con}$ and $\mathbf{t}_{digital}$, denoted as $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$.

$$\begin{bmatrix} \mathbf{Y}_{1,\mathbf{t}_{digital}} & \mathbf{Y}_{1,\mathbf{t}_{con}} \\ \mathbf{Y}_{2,\mathbf{t}_{digital}} & \mathbf{Y}_{2,\mathbf{t}_{con}} \\ \vdots & \vdots \\ \mathbf{Y}_{N_{cultivar},\mathbf{t}_{digital}} & \mathbf{Y}_{N_{cultivar},\mathbf{t}_{con}} \end{bmatrix} = \mathbf{I}_{N_{cultivar}} \times \begin{bmatrix} u_{1,\mathbf{t}_{digital}} & u_{1,\mathbf{t}_{con}} \\ u_{2,\mathbf{t}_{digital}} & u_{2,\mathbf{t}_{con}} \\ \vdots & \vdots \\ u_{N_{cultivar},\mathbf{t}_{digital}} & u_{N_{cultivar},\mathbf{t}_{con}} \end{bmatrix} + \begin{bmatrix} \varepsilon_{1,\mathbf{t}_{digital}} & \varepsilon_{1,\mathbf{t}_{con}} \\ \varepsilon_{2,\mathbf{t}_{digital}} & \varepsilon_{2,\mathbf{t}_{con}} \\ \vdots & \vdots \\ \varepsilon_{N_{cultivar},\mathbf{t}_{digital}} & \varepsilon_{N_{cultivar},\mathbf{t}_{con}} \end{bmatrix}. \tag{7.1}$$

We compared estimated genetic correlations calculated by the multi-trait model $\hat{\rho}^{Gen}_{multi-trait}$ and the bi-trait model $\hat{\rho}^{Gen}_{bi-trait}$ in Figures 7.1 and 7.2. $\hat{\rho}^{Gen}_{multi-trait}$ and $\hat{\rho}^{Gen}_{bi-trait}$ showed minor differences in the heatmap and PDF, indicating similar results for $\hat{\rho}^{Gen}$. The multi-trait model took 145.709 s and the bi-trait model took 2260.547 s. We chose the bi-trait model for two reasons. The dataset with 315 cultivars is relatively small so the bi-trait model could compute all $\hat{\rho}^{Gen}_{bi-trait}$s in a reasonable time. Besides, we would choose from models with and without fixed effects for each trait pair in Section 7.2. The bi-trait model provided enough flexibility.

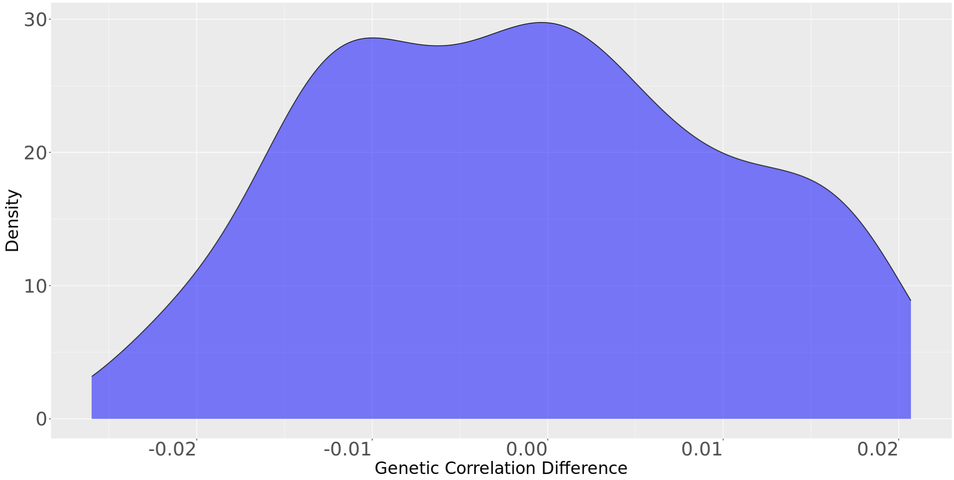


Figure 7.1: The estimated PDF of $(\hat{\rho}^{Gen}_{multi-trait} - \hat{\rho}^{Gen}_{bi-trait})$ by Gaussian kernel density

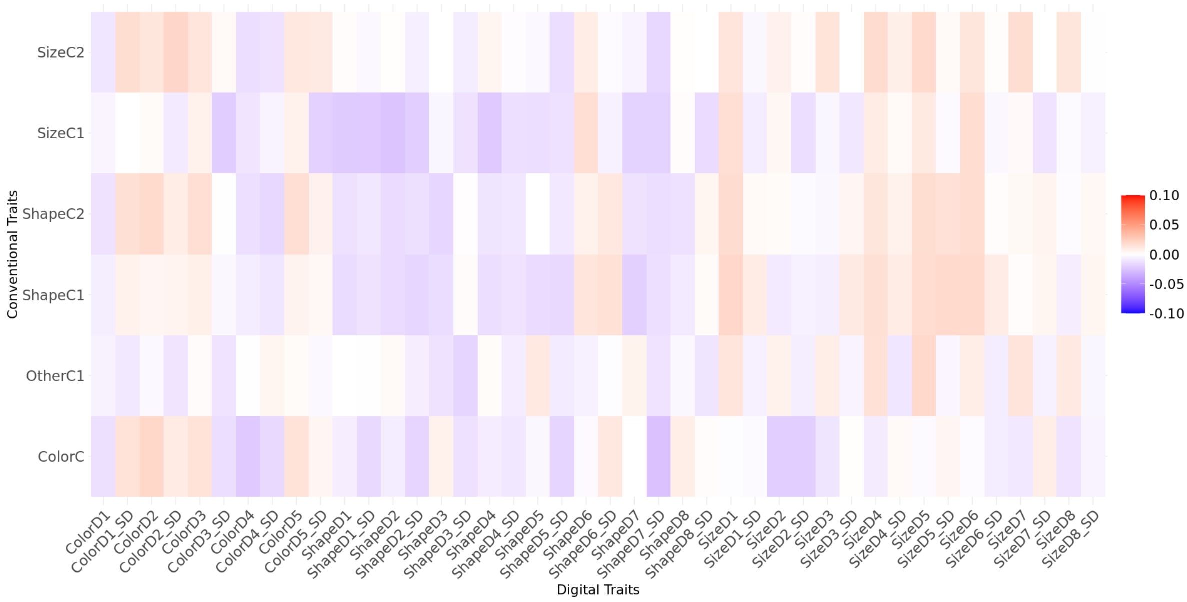


Figure 7.2: The heatmap of $(\hat{\rho}^{Gen}_{multi-trait} - \hat{\rho}^{Gen}_{bi-trait})$

The *multitrait()* function applies the Gibbs sampler to estimate posterior distributions, where the burn-in iterations hyperparameter $N_{iterations}$ is crucial for reliability. We presented plots of $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ with varying $N_{iterations}$ from 501 to 1500 in Figure 7.3 using a bi-trait model without fixed effects. $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ varied significantly at low $N_{iterations}$. As $N_{iterations}$ increased, $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ oscillated around a constant. Figure 7.3 showed that beyond 1000 iterations, the oscillation amplitude of $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ stabilized. Similar plots for all $(\mathbf{t}_{con}, \mathbf{t}_{digital})$ pairs suggest using the first 1000 iterations as burn-in for all. The mean genetic correlation from iterations 1001st to 1100th was $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$.

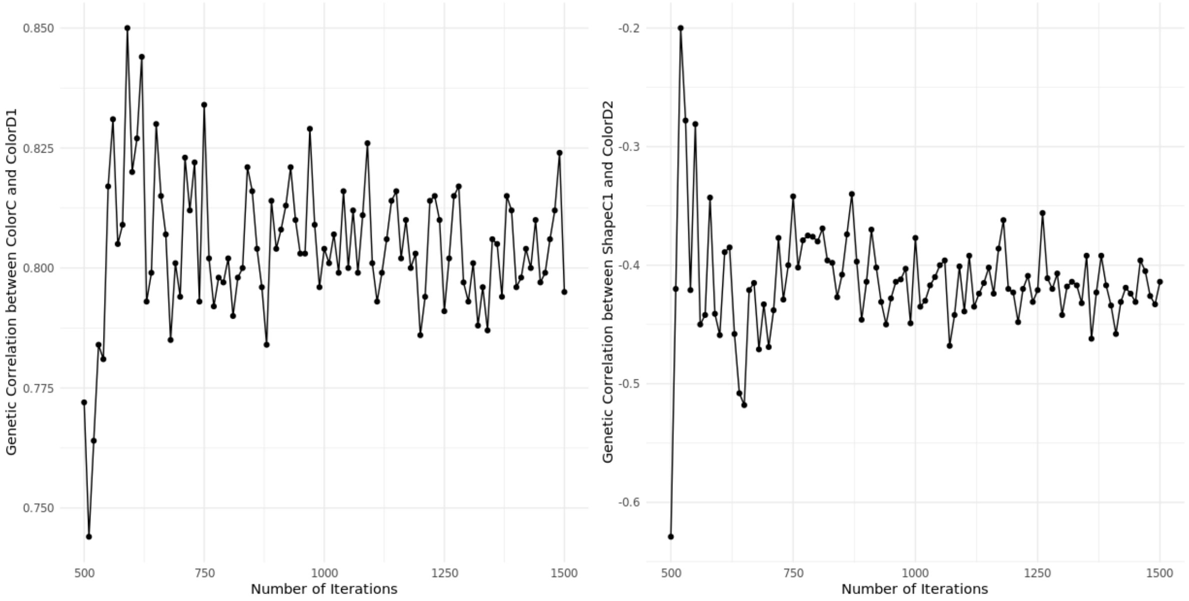


Figure 7.3: Sensitivity analysis of $N_{iterations}$

7.2. Fruit Colors as Fixed Effects

Fruit color is an important feature of blocky peppers. It may have an impact on the calculation of $\hat{\rho}^{Gen}$ because cultivars of the same color are more likely to have a stronger genetic relationship. The blocky pepper has three colors: red, yellow, and orange. So, the color variables *red*, *yellow* and *orange* are candidates for the fixed effect in the bi-trait model. Table 7.1 shows the numbers of cultivars in each color. More than half of the cultivars are red and orange is the rarest color.

Color	Orange	Yellow	Red	Total
Number of Cultivars	50	84	181	315

Table 7.1: Size of the sub-sample of cultivars for each color

We compared two bi-trait models: a full model with fixed effects $\mathbf{Y} = \mu + \mathbf{X}\beta + \mathbf{Z}u$ and a nested model without fixed effects $\mathbf{Y} = \mu + \mathbf{Z}u$, visualizing the absolute values of $\hat{\rho}^{Gen}$s in Figure 7.4. The results indicated that $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ could differ between models; for example, absolute $\hat{\rho}^{Gen}_{\text{ColorC,ColorD1}}$ was less than 0.5 with the full model but exceeded 0.5 with the nested model.

The likelihood ratio test determines if adding fixed effects improves the model compared to the nested model [9]. It is expressed as

$$\mathcal{H}_0: \text{The nested model out performances the full model.}$$
$$\text{against } \mathcal{H}_1: \text{The full model out performances the nested model.}$$

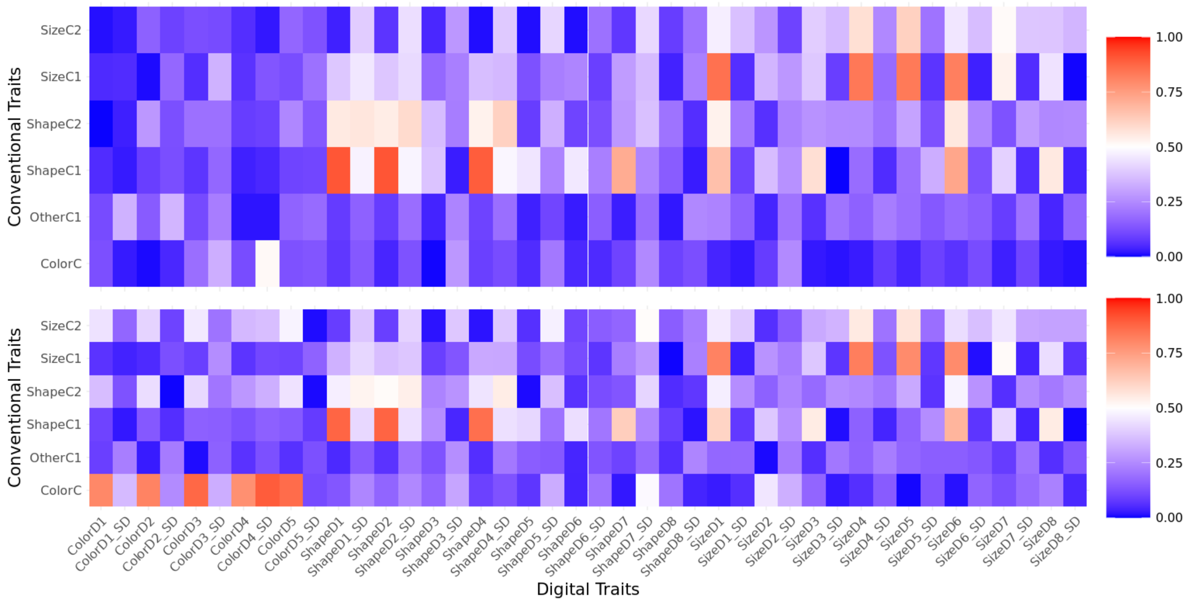


Figure 7.4: Heatmaps of Abs($\hat{\rho}^{Gen}$)s by bi-trait models
Up: Full model (with fixed effects); Down: Nest model (without fixed effects)

The likelihood-ratio test is based on the difference between log-likelihoods:

$$-2[\text{loglikelihood(nested model) - loglikelihood(full model)}].$$

Figure 7.5 visualized the P-values and selected models. It suggest that the full model may performed better than the nested model when at least one of $\mathbf{t}_{con}$ and $\mathbf{t}_{digital}$ was a color trait. If no color traits were involved, the nested model always outperformed. Recall the second reason to choose the bi-trait model mentioned in Section 7.1. We can only use either $\mathbf{Y} = \mu + \mathbf{X}\beta + \mathbf{Z}u$ or $\mathbf{Y} = \mu + \mathbf{Z}u$ for all trait pairs if we chose the multi-trait model with all pairs.

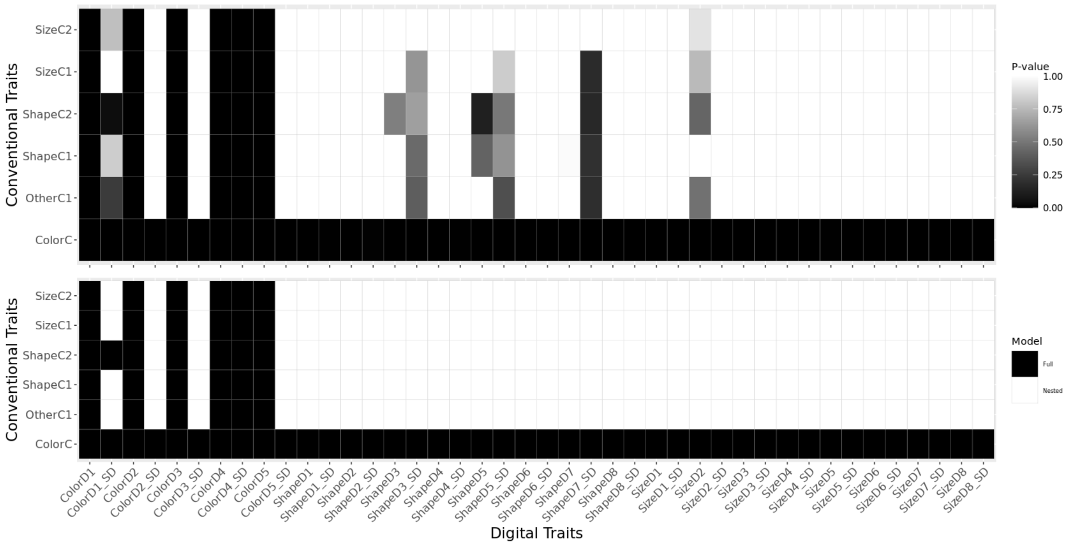


Figure 7.5: Heatmaps of model selection
Up: P-values of likelihood ratio test; Down: Selected models with threshold P-value = 0.05.

Figure 7.6 presented the heatmap of $\hat{\rho}^{Gen}$s from the model chosen by the likelihood ratio test. They would be used in variable selection next. Alongside estimated genetic correlation $\hat{\rho}^{Gen}$, estimated Pearson correlation $\hat{\rho}^{Pearson}$ is another common correlation coefficient. Figure 7.7 compared the coefficients by visualizing $(\hat{\rho}^{Pearson} - \hat{\rho}^{Gen})$. A significant difference between $\hat{\rho}^{Gen}$ and $\hat{\rho}^{Pearson}$ appeared when $\mathbf{t}_{con}$ and $\mathbf{t}_{digital}$ were both color traits.

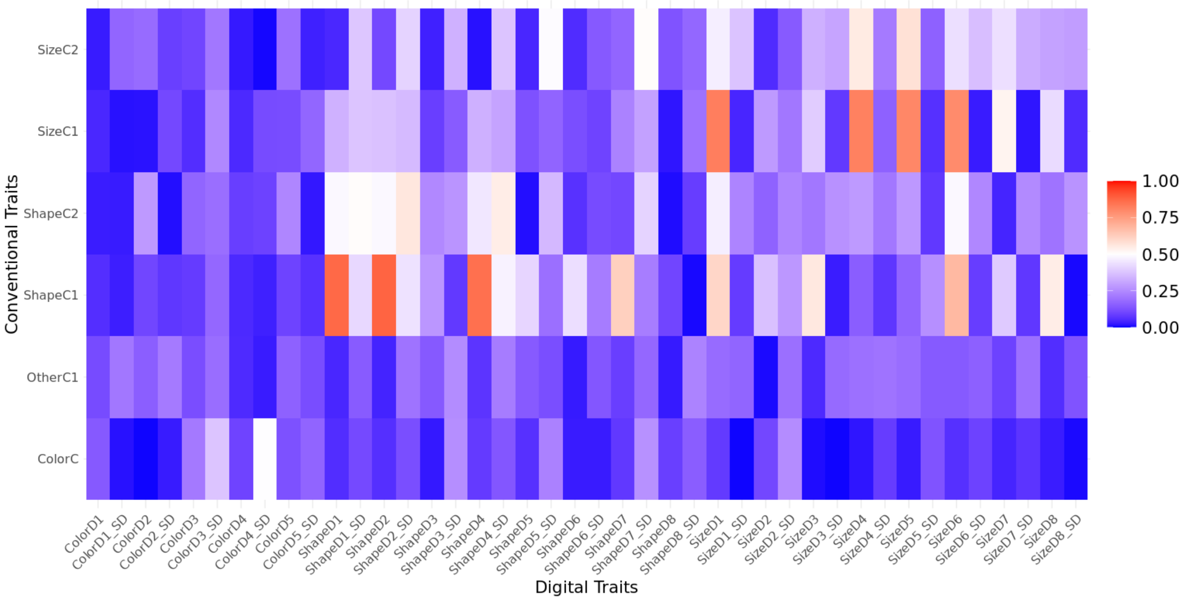


Figure 7.6: Heatmap of Abs($\hat{\rho}^{Gen}$)s from the model selected by likelihood ratio test

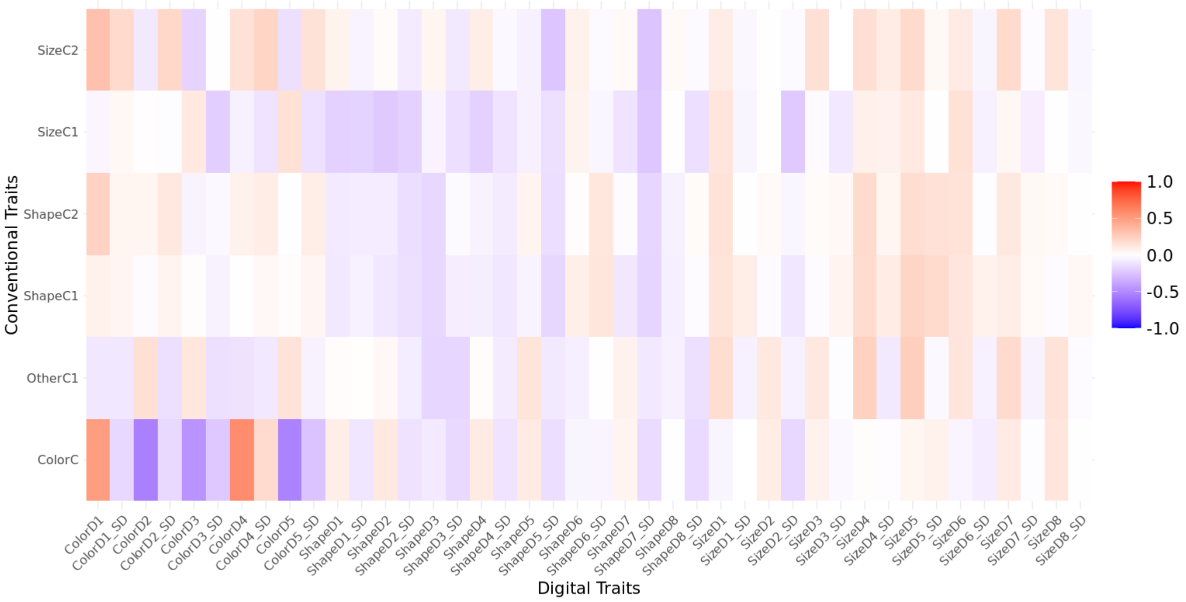


Figure 7.7: Heatmap of ($\hat{\rho}^{Pearson}$-$\hat{\rho}^{Gen}$)s

7.3. Distributions of the Estimated Genetic Correlation $\hat{\rho}^{Gen}$

After abandoning the conventional noninformative trait *OtherC2*, 42 digital and six conventional traits remain, which lead to (42 × 6 =) 252 $\hat{\rho}^{Gen}$s. Figure 7.8 displayed the histogram of absolute $\hat{\rho}^{Gen}$s, with more than half ($\mathbf{t}_{con}, \mathbf{t}_{digital}$) of the pairs showing no correlation, 88 pairs having intermediate correlations, and six pairs being highly correlated.

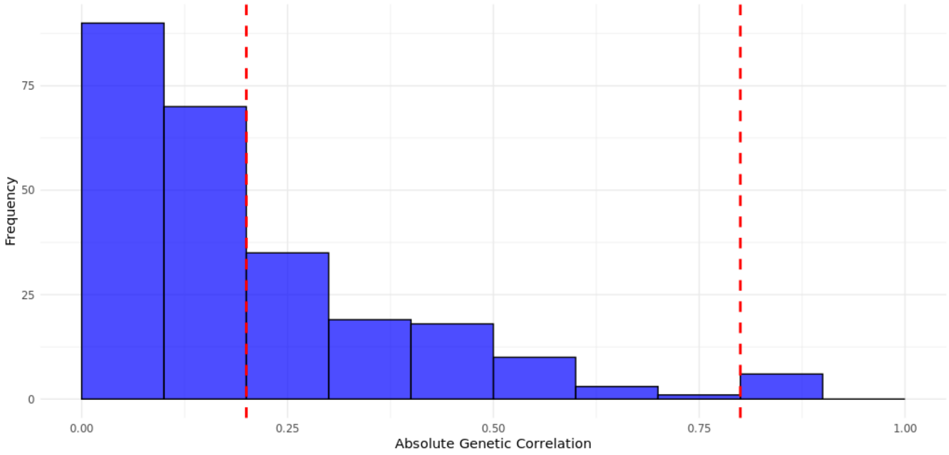


Figure 7.8: Histogram of absolute $\hat{\rho}^{Gen}$s
0.2 and 0.8 are the boundary values for $\hat{\rho}^{Gen}$ groups (low/intermediate/high).

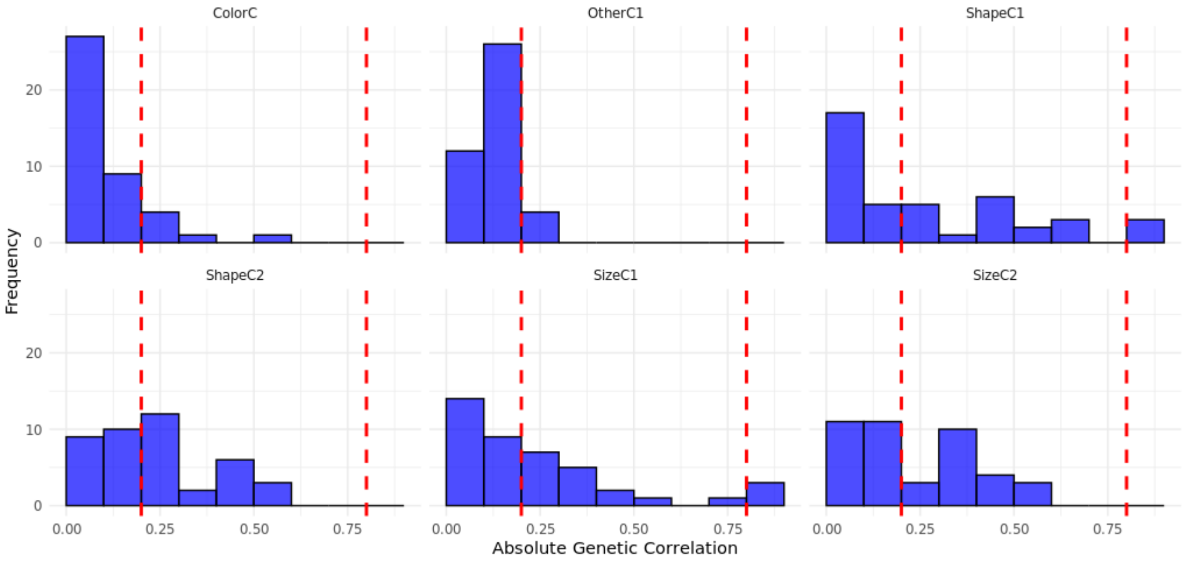


Figure 7.9: Histograms of absolute $\hat{\rho}^{Gen}$s by conventional trait
0.2 and 0.8 are the boundary values for $\hat{\rho}^{Gen}$ groups (low/intermediate/high).

We displayed distributions of abs($\hat{\rho}^{Gen}$)s of each conventional trait in Figure 7.9 and Table 7.2. Two conventional traits, *ShapeC1* and *SizeC1*, could be replaced by correlated digital traits, listed in Table 7.3. Trait *ShapeC1* can be replaced by digital shape traits *ShapeD1*, *ShapeD2*, *ShapeD4*, and digital size traits *SizeD1*, *SizeD4*, *SizeD5*. This is supported by the heatmap in Figure 6.2, showing within-group correlations (color, shape, size) are stronger than between-group. Breeders will select digital traits to replace *ShapeC1* and *SizeC1* based on cost and interpretability.

$\mathbf{t}_{con}$	The number of $\mathbf{t}_{digital}$s with low $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$	The number of $\mathbf{t}_{digital}$s with intermediate $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$	The number of $\mathbf{t}_{digital}$s with high $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$
ColorC	35	7	0
OtherC	37	5	0
ShapeC1	22	17	3
ShapeC2	18	24	0
SizeC1	24	15	3
SizeC2	22	20	0

Table 7.2: Repartition of the digital traits per correlation group (low/intermediate/high) for each conventional trait.

Conventional Trait $\mathbf{t}_{con}$	Digital Trait $\mathbf{t}_{digital}$	$\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$	$\hat{\rho}^{Pearson}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$
ShapeC1	ShapeD1	0.873	0.777
	ShapeD2	0.883	0.782
	ShapeD4	0.857	0.784
SizeC1	SizeD1	-0.825	-0.686
	SizeD4	-0.820	-0.743
	SizeD5	-0.806	-0.688

Table 7.3: The pairs with high $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ and their estimated correlations

8

Prediction of Conventional Traits

We move to the 'Statistical Learning Prediction' phase of the workflow (Figure 1.1). Initially, there were seven conventional traits; *OtherC2* was discarded, while *ShapeC1* and *SizeC1* were replaced with digital traits. The goal in this chapter is to predict traits *ColorC*, *ShapeC2*, *SizeC2*, and *OtherC1* using linear regression, LASSO regression, random forest, and XGBoost. All conventional and digital trait values were centered in this chapter.

8.1. The Role of Variable Selection by $\hat{\rho}^{Gen}$

To investigate the role of variable selection by $\hat{\rho}^{Gen}$ on conventional trait prediction, we tested two predictor sets: $\mathbf{T}_{digital}$ and $\mathbf{T}_{digital,\mathbf{t}_{con}}$. The first set contained all digital traits $\mathbf{T}_{digital}$, while the second set $\mathbf{T}_{digital,\mathbf{t}_{con}}$ included digital traits meeting the criterion $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}} \geq 0.2$, leading to $\mathbf{T}_{digital,\mathbf{t}_{con}} \subset \mathbf{T}_{digital}$. The conventional trait value is $\mathbf{Y}_{cultivar,\mathbf{t}_{con}}$, and the predicted value is $\hat{\mathbf{Y}}^{Model,Predictor\ Set}_{cultivar,\mathbf{t}_{con}}$ given $Model \in \{Linear\ regression, LASSO\ regression, Random\ forest, XGBoost\}$, and $Predictor\ Set \in \{\mathbf{T}_{digital}, \mathbf{T}_{digital,\mathbf{t}_{con}}\}$. We used mean square error (MSE) and Pearson correlation between real and predicted values $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ to evaluate model performance. MSE is given by

$$\frac{1}{N_{cultivar}} \sum_{i=1}^{N_{cultivar}} (\mathbf{Y}_{i,\mathbf{t}_{con}} - \hat{\mathbf{Y}}^{Model,Predictor\ Set}_{i,\mathbf{t}_{con}})^2,$$

and Pearson correlation between real and predicted values is defined by the

$$\frac{\sum_{i=1}^{N_{cultivars}}(\mathbf{Y}_{i,\mathbf{t}_{con}} - \bar{\mathbf{Y}}_{\mathbf{t}_{con}})(\hat{\mathbf{Y}}_{i,\mathbf{t}_{con}}^{Model,Predictor\ Set} - \bar{\hat{\mathbf{Y}}}_{\mathbf{t}_{con}}^{Model,Predictor\ Set})}{\sqrt{\sum_{i=1}^{N_{cultivars}}(\mathbf{Y}_{i,\mathbf{t}_{con}} - \bar{\mathbf{Y}}_{\mathbf{t}_{con}})^2 \sum_{i=1}^{N_{cultivars}}(\hat{\mathbf{Y}}_{i,\mathbf{t}_{con}}^{Model,Predictor\ Set} - \bar{\hat{\mathbf{Y}}}_{\mathbf{t}_{con}}^{Model,Predictor\ Set})^2}}.$$

5-fold cross-validation splits the dataset into five equal subsets. In each iteration, one subset serves as the validation set, and the rest are for training. This process repeats five times, with each subset used once for validation. The final metric is the average of all iterations. [56].

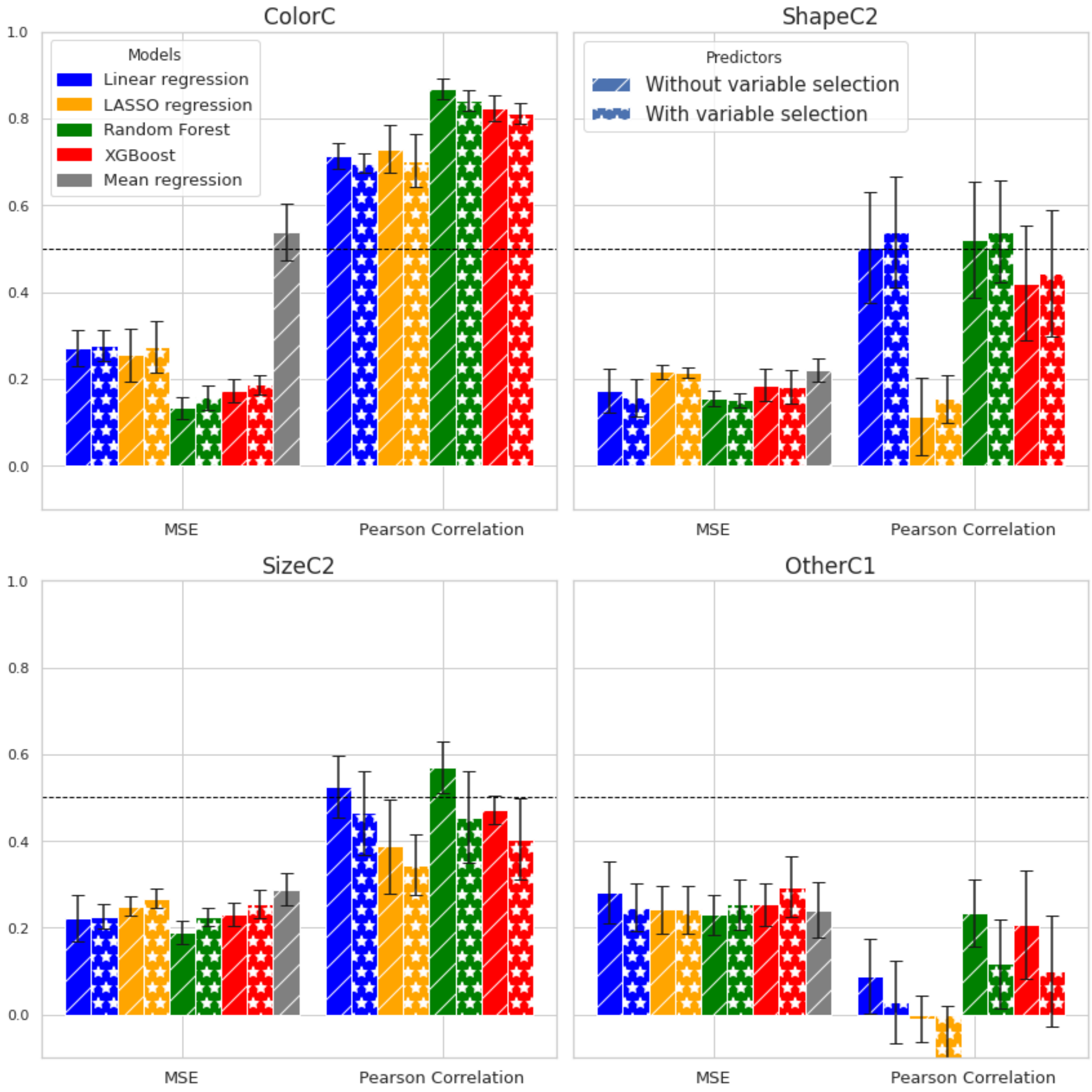


Figure 8.1: The performance of predictive models on $\mathbf{t}_{con}$ and $\mathbf{T}_{digital,\mathbf{t}_{con}}$

To compare model performance on $\mathbf{T}_{digital}$ and $\mathbf{T}_{digital,\mathbf{t}_{con}}$, we illustrated the MSE and $\hat{\rho}_{\mathbf{Y},\hat{\mathbf{Y}}}^{Pearson}$ across predictor sets in Figure 8.1, with error bars indicating the standard deviation of MSE. We included mean regression as a baseline model, using the average of conventional trait values in training set to predict the same trait in test set.

Figure 8.1 showed that MSE differences and $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ were associated with $\mathbf{t}_{con}$. The model underperformed on $\mathbf{T}_{digital,\mathbf{t}_{con}}$ predicting *ColorC*, *SizeC2*, and *OtherC1*, while *ShapeC2* predictions were more accurate on $\mathbf{T}_{digital,\mathbf{t}_{con}}$ than on $\mathbf{T}_{digital}$.

Model performance differences were MSE($\mathbf{T}_{digital,\mathbf{t}_{con}}$) - MSE($\mathbf{T}_{digital}$) and $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}(\mathbf{T}_{digital,\mathbf{t}_{con}})-\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}(\mathbf{T}_{digital})$. We used the Gaussian kernel density estimator to estimate the probability densities of these differences in Figure 8.2.

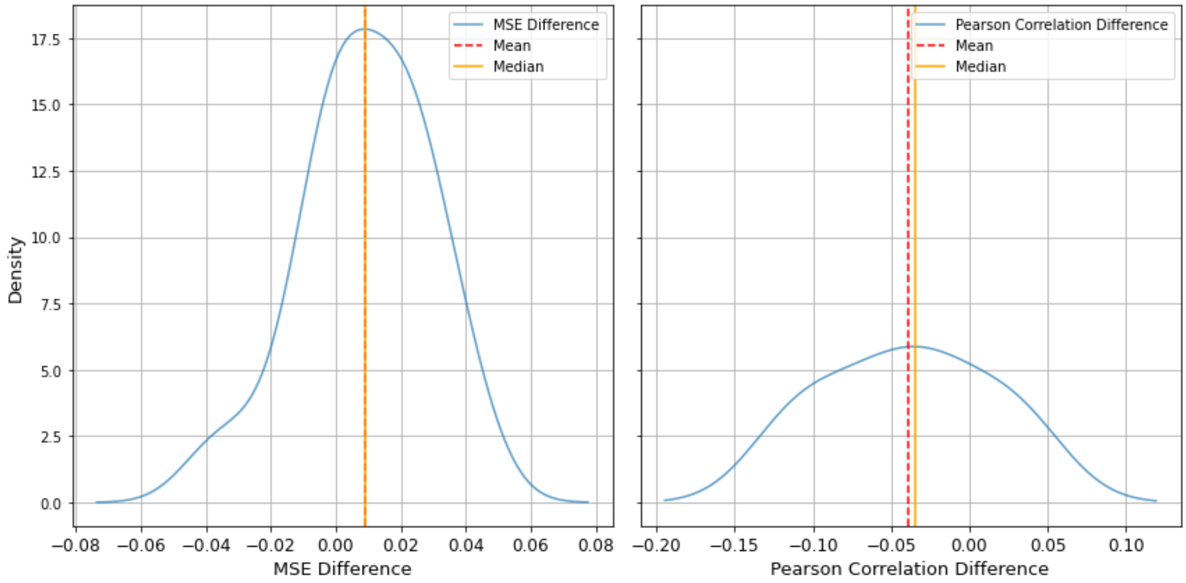


Figure 8.2: The estimated distributions of MSE difference and $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ difference
Left: MSE($\mathbf{T}_{digital,\mathbf{t}_{con}}$) - MSE($\mathbf{T}_{digital}$); Right: $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}(\mathbf{T}_{digital,\mathbf{t}_{con}}) - \hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}(\mathbf{T}_{digital})$

The means of the differences were very close to zero, so we used the two-sample t-test to determine if the differences were significant. We did the hypothesis test:

$$\mathcal{H}_0 : \mathbb{E}[\text{MSE}(\mathbf{T}_{digital,\mathbf{t}_{con}})] = \mathbb{E}[\text{MSE}(\mathbf{T}_{digital})].$$
$$\text{against } \mathcal{H}_1 : \mathbb{E}[\text{MSE}(\mathbf{T}_{digital,\mathbf{t}_{con}})] \neq \mathbb{E}[\text{MSE}(\mathbf{T}_{digital})].$$

We were also interested in the hypothesis test:

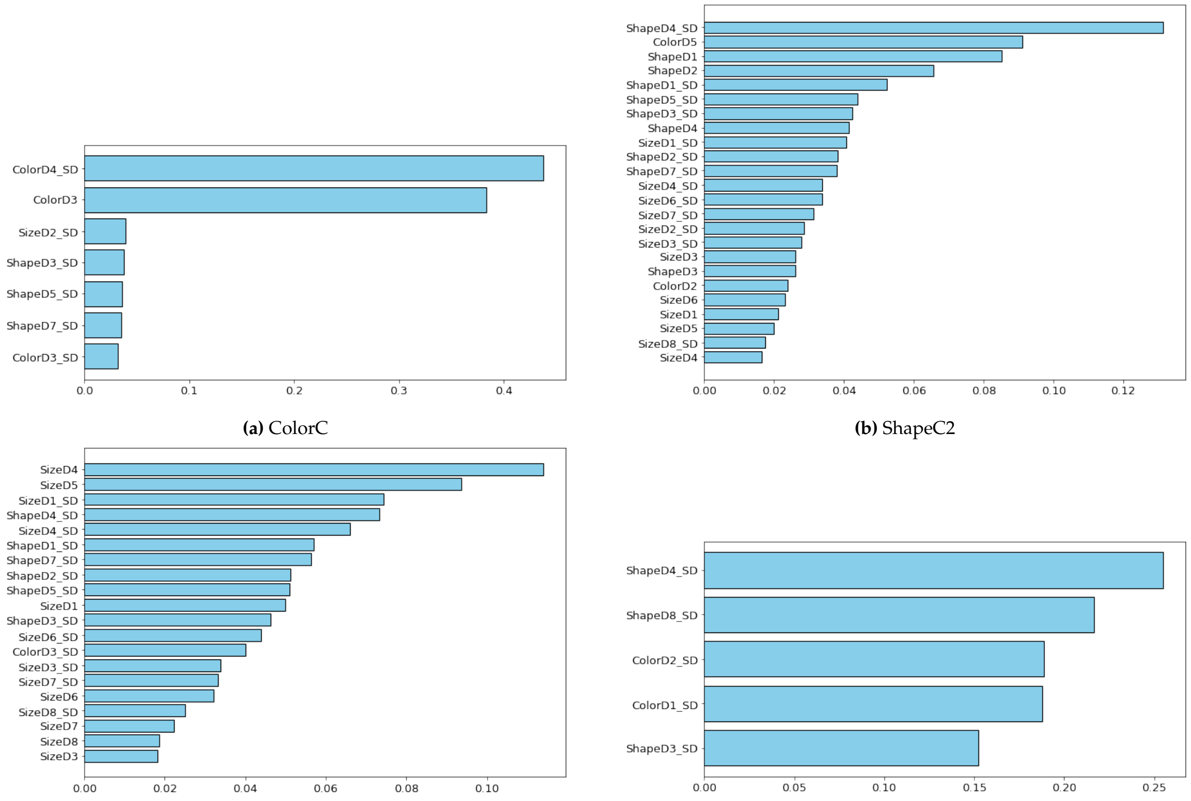
$$\mathcal{H}_0 : \mathbb{E}[\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}(\mathbf{T}_{digital,\mathbf{t}_{con}})] = \mathbb{E}[\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}(\mathbf{T}_{digital})].$$
$$\text{against } \mathcal{H}_1 : \mathbb{E}[\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}(\mathbf{T}_{digital,\mathbf{t}_{con}})] \neq \mathbb{E}[\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}(\mathbf{T}_{digital})].$$

The P-values, 0.57 and 0.69, indicated insignificant changes in MSE and $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$. Thus, $\mathbf{T}_{digital,\mathbf{t}_{con}}$ was chosen as the predictor set, simplifying the model without performance loss.

8.2. Important Features

Figure 8.3 presents the feature importance of each conventional traits by random forest. Each conventional trait had one or two key predictors: *ColorD4_SD* and *ColorD3* for *ColorC*, *ShapeD4_SD* for *ShapeC2*, *SizeD4* for *SizeC2*, *ShapeD4_SD* and *ShapeD8_SD* for *OtherC1*. These traits and predictors belonged to the same group (Color, Shape, Size) and showed high Pearson correlations in the heatmap (Figure 6.2).

LASSO regression, like random forests, is able to selects important features using a penalty term $L1$. Setting λ to one, the important predictors for *ColorC* were *ColorD3* and *ColorD4_SD*, for *ShapeC2* and *SizeC2* were *ColorD5* and *SizeD4*, respectively, with none for *OtherC1*. Overall, this aligned with the random forest results.



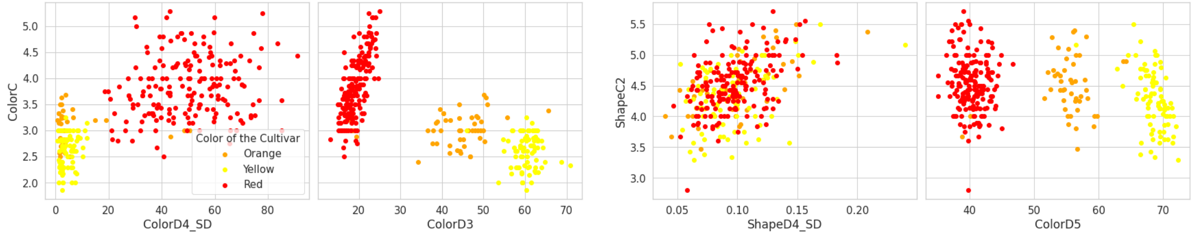
(a) ColorC

(b) ShapeC2

(c) SizeC2

(d) OtherC1

Figure 8.3: Feature importance from decrease in impurity

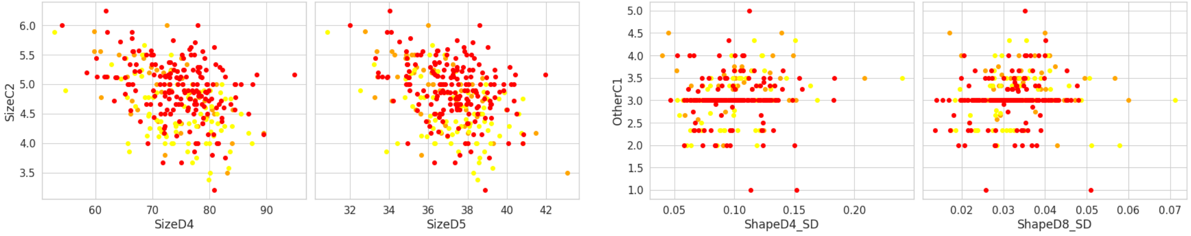


(a) ColorC

(b) ShapeC2

Figure 8.4: Scatter plots of each $\mathbf{t}_{con}$ and its two most important $\mathbf{t}_{digital}$s (One)

Figures 8.4 and 8.5 displayed scatter plots between each conventional trait and its two most important predictors. It showed clustering in ColorC vs. ColorD3 and ShapeC2 vs. ColorD5: cultivars in the same color would cluster together on color traits.



(a) SizeC2

(b) OtherC1

Figure 8.5: Scatter plots of each $\mathbf{t}_{con}$ and its two most important $\mathbf{t}_{digital}$s (Two)

8.3. Hyperparameters Tuning

We tuned hyperparameters for LASSO regression, random forest, and XGBoost in this section. Scatter plots of linear regression showed centralized values of real and predicted conventional traits in Figure 8.6. Two fuzzy clusters in subplot *ColorC* indicated poor group distinction of linear regression in *ColorC*. In *OtherC1*, predictions were near the mean. Large training set errors indicated underfitting across all traits.

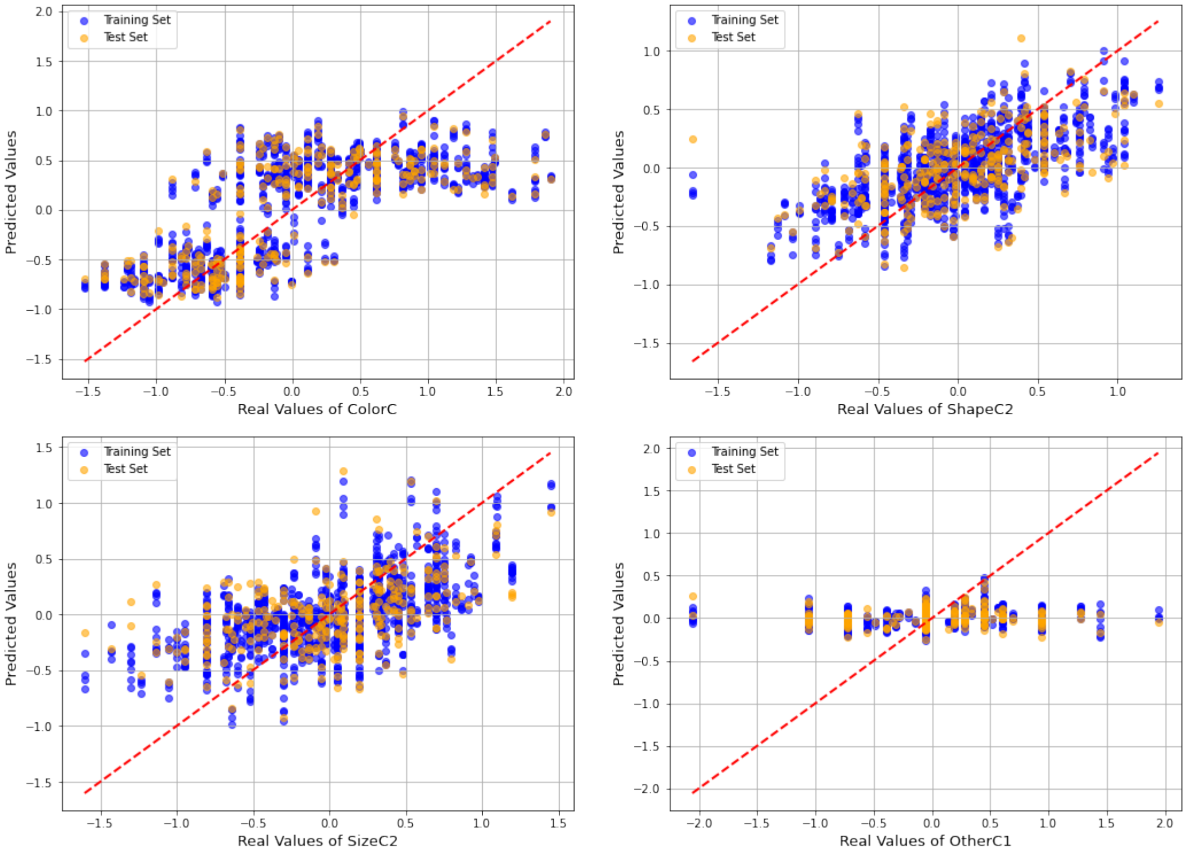


Figure 8.6: Scatter plots of real and predicted values by linear regression

8.3.1. Hyperparameters Tuning for LASSO Regression

The regularization parameter λ is the only hyperparameter in LASSO regression. To find the optimal λ, we conducted a sensitivity analysis of λ. λ values were fifty logarithmically spaced numbers between 10^{-2} and 10^{2}, plus zero. MSE changes with λ were shown in the first subplots of Figures 8.7 to 8.10. An increase in λ did not reduce MSE. MSE curves for *ColorC*, *ShapeC2*, and *SizeC2* formed an 'S' shape; MSE surged with increasing λ before stabilizing. For *OtherC2*, MSE stayed around 0.24 regardless of λ. Linear regression is a LASSO variant with $\lambda = 0$. Increasing λ did not enhance performance, aligning with Figure 8.1.

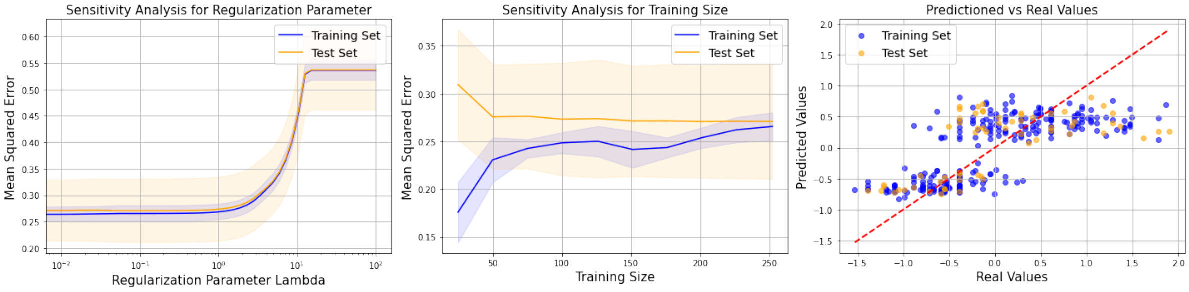


Figure 8.7: LASSO regression plots for ColorC

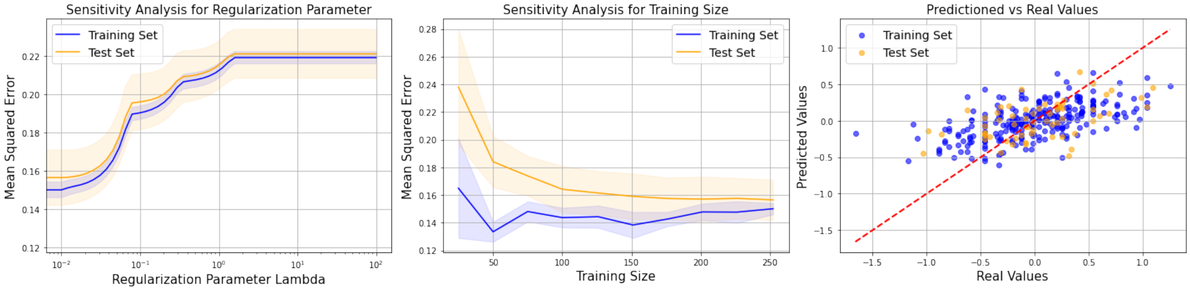


Figure 8.8: LASSO regression plots for ShapeC2

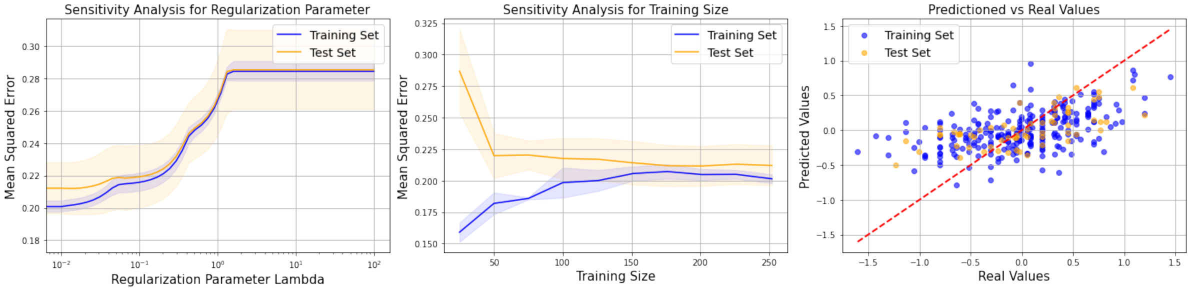


Figure 8.9: LASSO regression plots for SizeC2

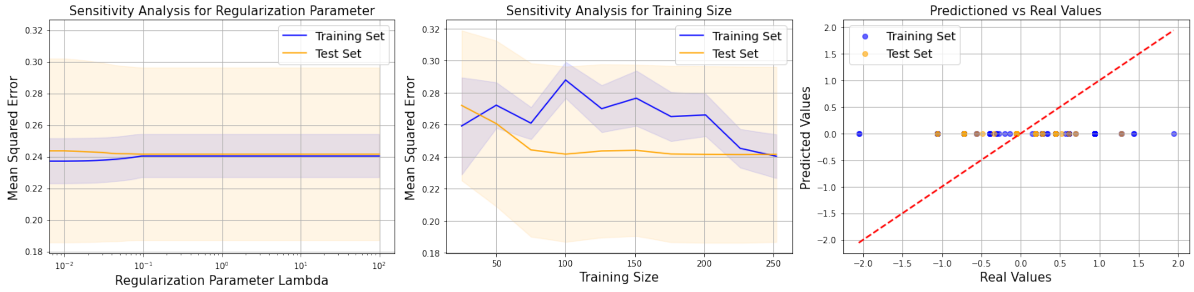


Figure 8.10: LASSO regression plots for OtherC1

For model fitting, we used two plots: MSE vs. training size and predicted vs. actual values scatter plot. A cross-validation generator spited the dataset into five training and test subsets. Using varying training sizes, we predicted conventional traits and calculated average MSE for each size [57]. The plots revealed that with sufficient training data, the MSE of training and test sets converged, indicating LASSO regression was not overfitted.

The scatter plot in Figure 8.7 showed that LASSO regression failed to distinguish clusters in *ColorC*. In Figure 8.10, LASSO predicted using the mean of training data, leading to an MSE identical to mean regression in Figure 8.1, explaining its weaker performance compared to linear regression on *OtherC1*. LASSO performed similarly on *ShapeC2* and *SizeC2*, consistent with Figure 8.1, indicating underfitting.

8.3.2. Hyperparameters Tuning for Random Forest and XGBoost

We focused on two key random forest hyperparameters: the number of trees N_{trees} and the minimum samples per leaf $N_{minsamples}$. Increasing $N_{minsamples}$ and decreasing N_{trees} can reduce model complexity, potentially lowering training performance but improving test performance. N_{trees} was an integer from 10 to 200, and $N_{minsamples}$ was an integer from 1 to 5. We selected the model with the lowest MSE across five cross-validations, then conducted a sensitivity analysis by varying N_{trees} or $N_{minsamples}$ while keeping other hyperparameters constant. Results were displayed in Figure 8.11 to Figure 8.14.

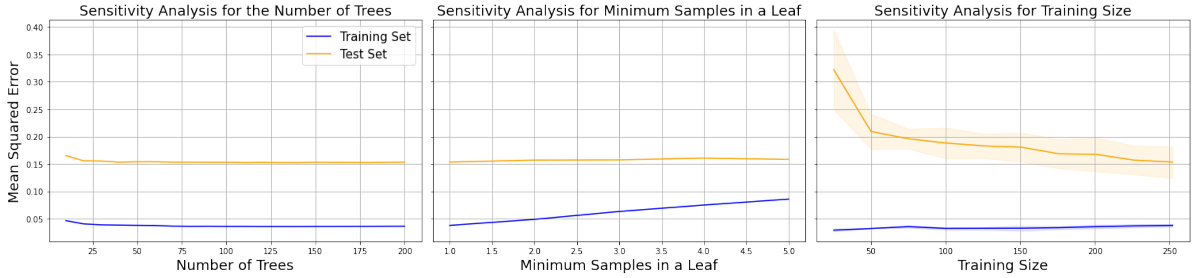


Figure 8.11: Sensitivity analysis of the random forest with two hyperparameters for ColorC

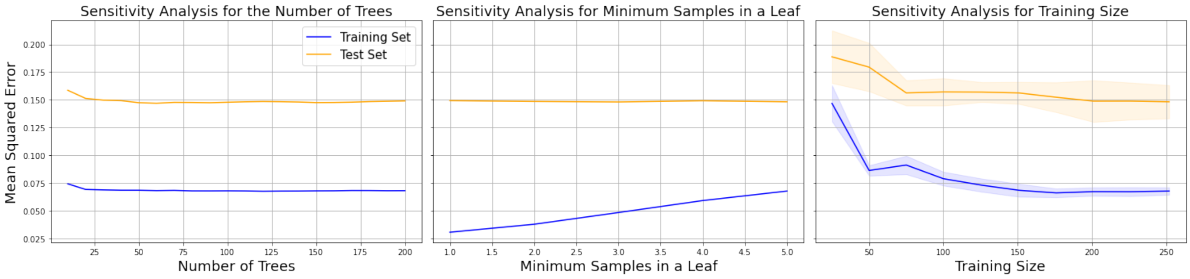


Figure 8.12: Sensitivity analysis of the random forest with two hyperparameters for ShapeC2

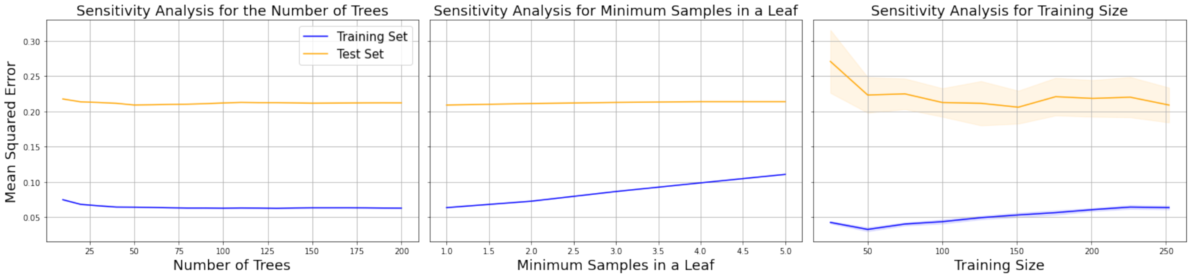


Figure 8.13: Sensitivity analysis of the random forest with two hyperparameters for SizeC2

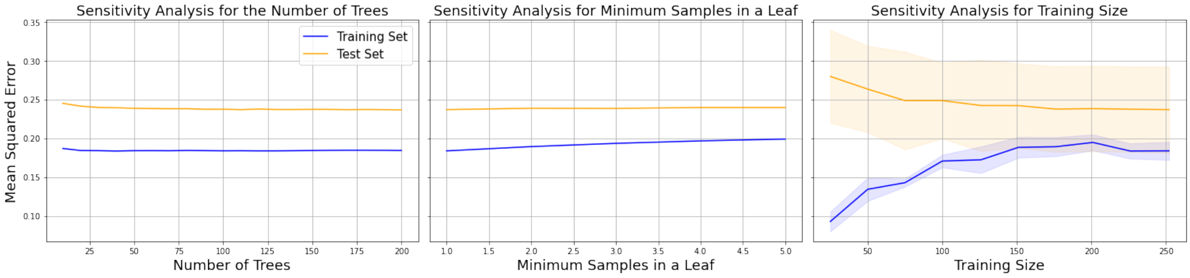


Figure 8.14: Sensitivity analysis of the random forest with two hyperparameters for OtherC1

Plots of all traits had a similar pattern: as N_{trees} increased, MSE for both test and training sets decreased initially, then stabilized. Training set MSE increased with $N_{minsamples}$, yet test set errors hardly decreased. Unexpectedly, the test set error for *OtherC1* rose with $N_{minsamples}$. The large MSE difference between training and test sets suggest potential overfitting of the model.

The sensitivity analysis of training size revealed that as training size grew, errors in both training and test sets initially decreased and then stabilized. MSE for both sets would converge to the same value if the model is well-fitted. However, the MSE converged to different values, indicating the random forest was overfitted.

Figure 8.15 presented scatter plots of predicted vs. actual values. As shown in *ColorC*, the random forest effectively distinguished *ColorC* due to the presence of a single group, outperforming linear and LASSO regressions on *ColorC*.

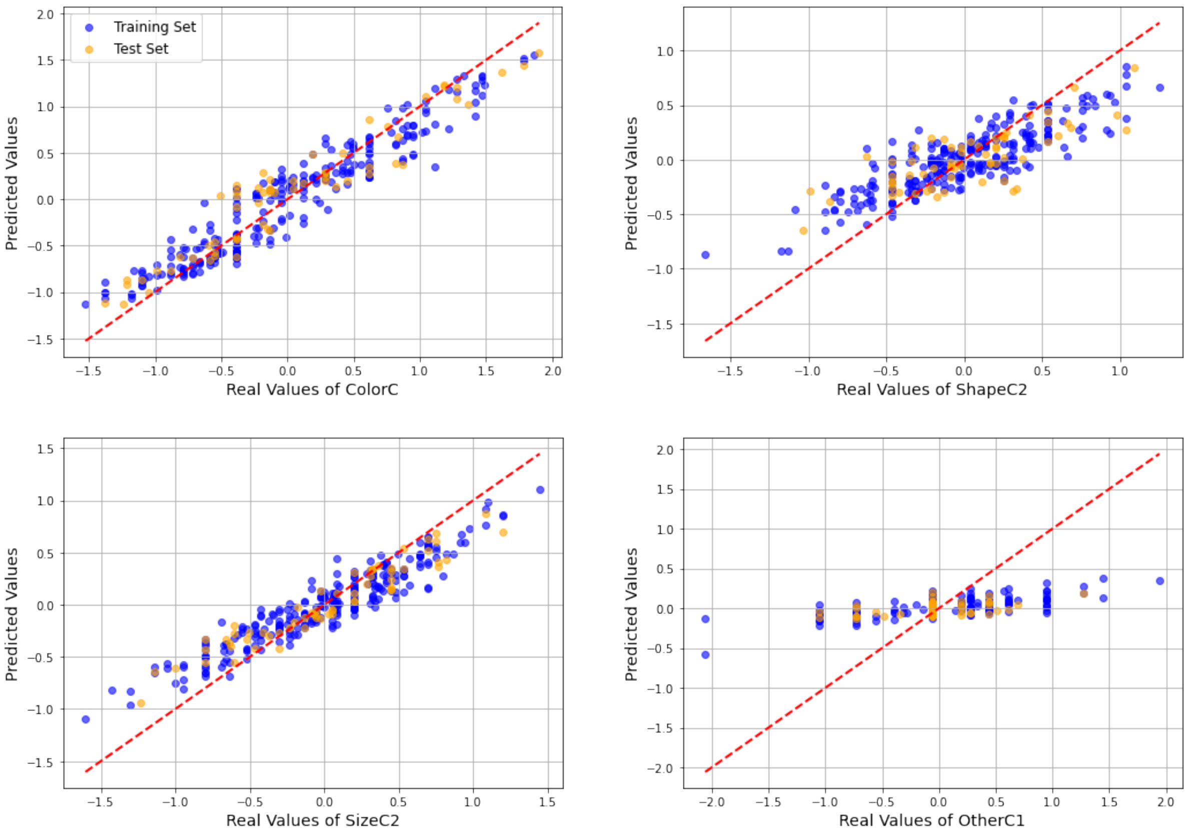


Figure 8.15: Scatter plots of real and predicted values by random forest

To address random forest overfitting, we included two more hyperparameters: max CART depth $N_{maxdepth}$ and feature proportion for splits $\frac{N_{subfeatures}}{N_{features}}$. To prevent overfitting, decreasing $N_{maxdepth}$ reduces CART complexity, and minimizing $\frac{N_{subfeatures}}{N_{features}}$ decreases correlations between CARTs. $N_{maxdepth}$ was an integer between three and ten, $\frac{N_{subfeatures}}{N_{features}}$ was between 0.1 and 0.7. We found the best model with the lowest MSE using five cross-validations across hyperparameters N_{trees}, $N_{minsamples}$, $N_{maxdepth}$ and $\frac{N_{subfeatures}}{N_{features}}$, then conducted sensitivity analysis on N_{trees}, $N_{minsamples}$ and training size.

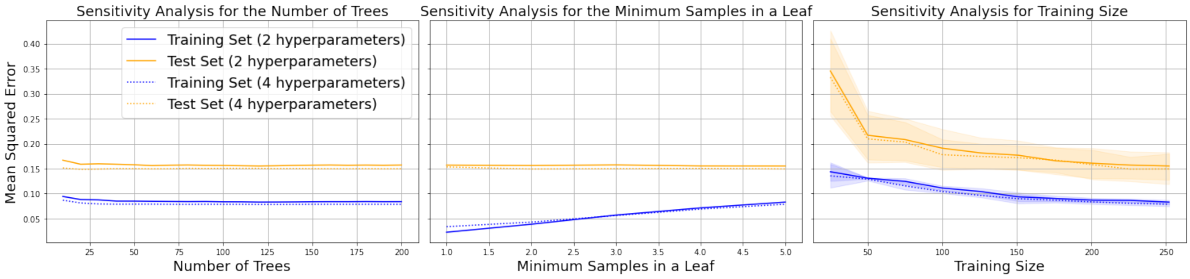


Figure 8.16: Sensitivity analysis of the random forest with four hyperparameters for ColorC

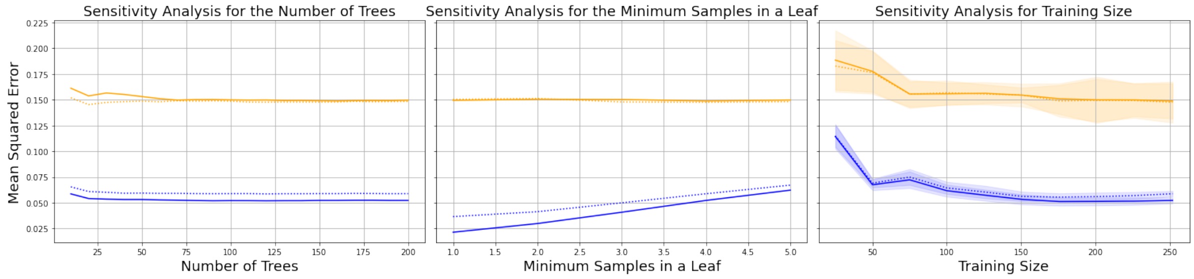


Figure 8.17: Sensitivity analysis of the random forest with four hyperparameters for ShapeC2

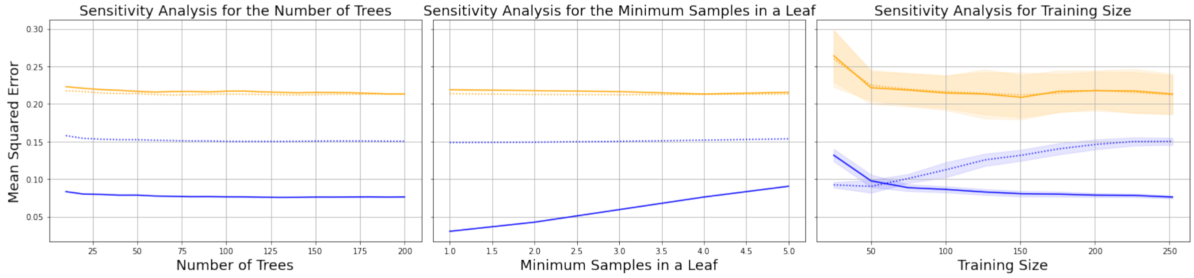


Figure 8.18: Sensitivity analysis of the random forest with four hyperparameters for SizeC2

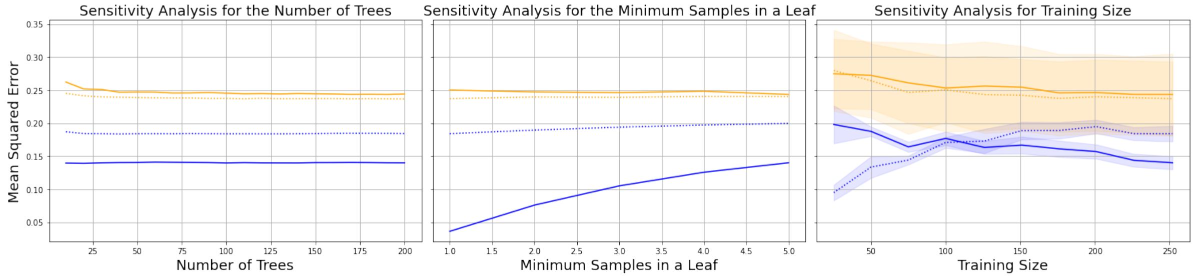


Figure 8.19: Sensitivity analysis of the random forest with four hyperparameters for OtherC1

Adding hyperparameters minimally affected the test set MSE, while greatly impacting the training set MSE. MSE decreased for the *ColorC* training set but increased for the other three conventional trait training sets. Adding hyperparameters did not prevent overfitting; the model with four hyperparameters remained overfitted. As training set size grew, the MSE for training and test sets failed to converge.

After tuning random forest hyperparameters, we tuned XGBoost on two sets: N_{trees}, $N_{minsamples}$ and N_{trees}, $N_{minsamples}$, $N_{maxdepth}$, $\frac{N_{subfeatures}}{N_{features}}$. We encountered the same problem as the random forest. The best XGBoost from N_{trees}, $N_{minsamples}$ was overfitted. Adjusting $N_{maxdepth}$ and $\frac{N_{subfeatures}}{N_{features}}$ did not solve the problem. Besides, XGBoost outperformed on the training set but had larger test errors than random forest, indicating more severe overfitting.

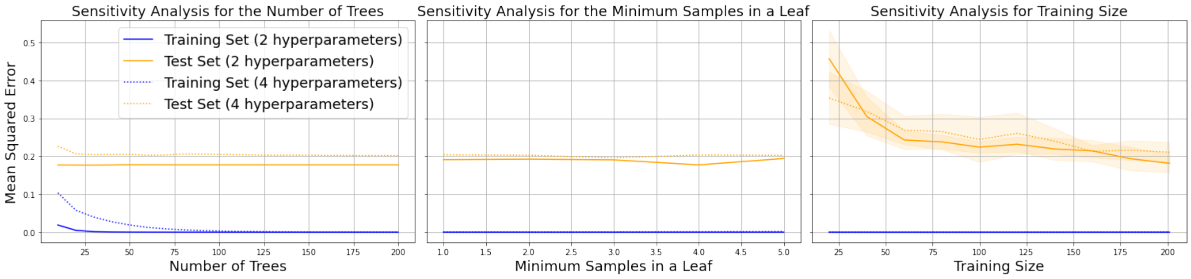


Figure 8.20: Sensitivity analysis and prediction results of the XGBoost for ColorC

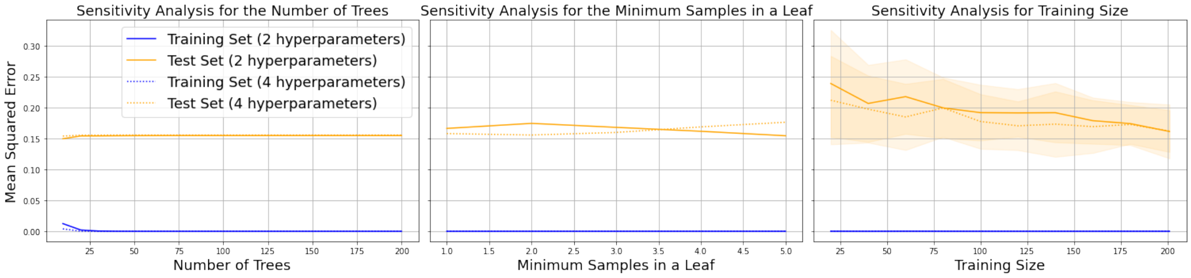


Figure 8.21: Sensitivity analysis and prediction results of the XGBoost for ShapeC2

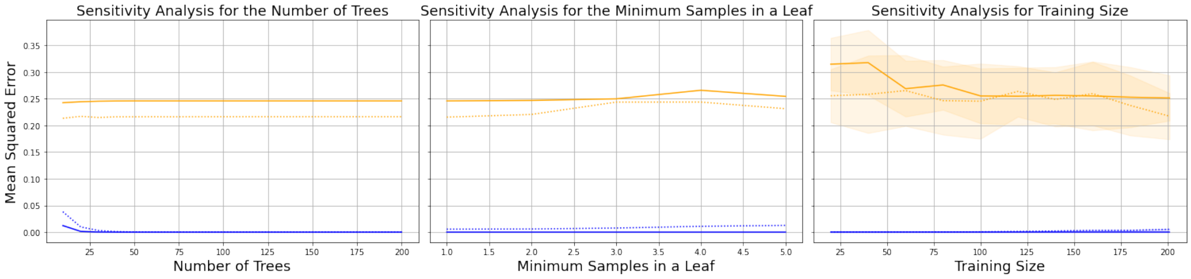


Figure 8.22: Sensitivity analysis and prediction results of the XGBoost for SizeC2

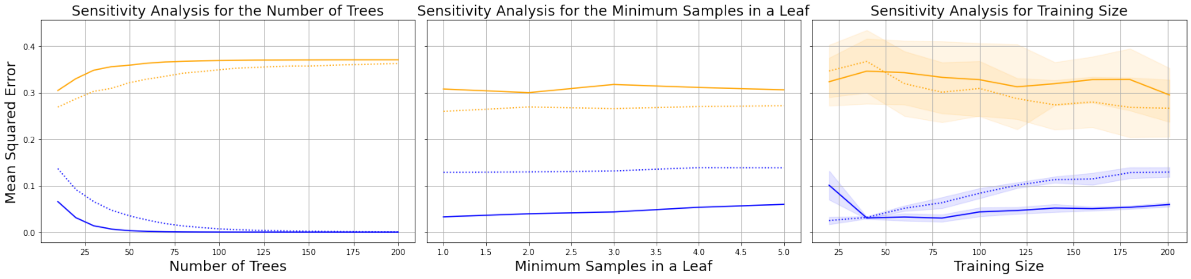


Figure 8.23: Sensitivity analysis and prediction results of the XGBoost for OtherC1

Figure 8.24 displayed scatter plots of real and predicted values from XGBoost. Training set points lied closer to the diagonal than test set points, indicating greater test errors. Comparing with random forest scatter plots in Figure 8.15, XGBoost was more overfitted. Although the tree-based models were overfitted, they still outperformed the underfitted linear and LASSO regressions models on test sets, showing lower MSE and higher $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ values in Figure 8.1.

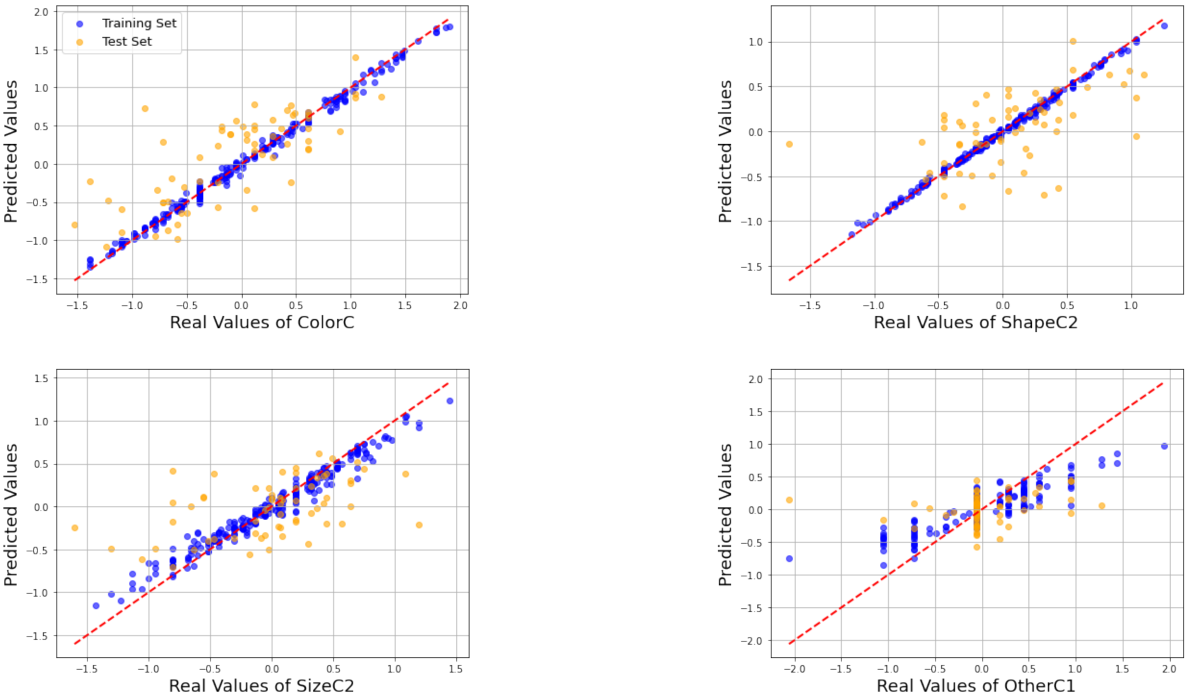


Figure 8.24: Scatter plots of real and predicted values by XGBoost

8.4. Comparison of Model Performance

The predictive models were evaluated after tuning, with performance displayed in Figure 8.25 and standard deviations from 5 cross-validation. A key threshold for breeders is $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ greater than 0.5. All models successfully predicted *ColorC* because their $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$s exceeded 0.5 significantly. Linear regression and random forest models could predict *ShapeC2* with $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$s between 0.5 and 0.6. Random forests could also predict *SizeC2* with $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$s over 0.5 slightly. No models could predict *OtherC1*.

Figure 8.25 showed the MSE of mean regression as a baseline, using the training set's mean values for predictions. For *ColorC*, linear and LASSO regression cut MSE by about 50%, while tree-based models reduced it by one-third. In *ShapeC2* and *SizeC2*, all models had slightly lower MSE than mean regression, but in *OtherC2*, their MSE were almost equal to it.

Additionally, it was observed that MSE and $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ did not always get the same rankings of model performance. They agreed on the predictable traits: *ColorC*, *ShapeC2*, and *SizeC2* that the random forest always had the best performance. But they differed in *OtherC1* because *OtherC1* was unpredictable. Since we tried a large amount of models this may be because there was not enough information in the predictors to predict *OtherC1*.

Linear regression and random forest have similar MSE to mean regression on *ShapeC2* and *SizeC2*, but $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ shows they can predict the traits, indicating $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.5$ is not a suitable threshold. Model performance on *ColorC* suggests $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.7$ as a better threshold. The model's MSE is significantly lower than mean regression's when $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}} > 0.7$. However, since we did not have time to further study the new threshold with breeders, this study still used $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.5$ as the threshold. Figure 8.26 showed that hyperparameter tuning minimally affected performance, reducing test set MSE by less than 0.1.

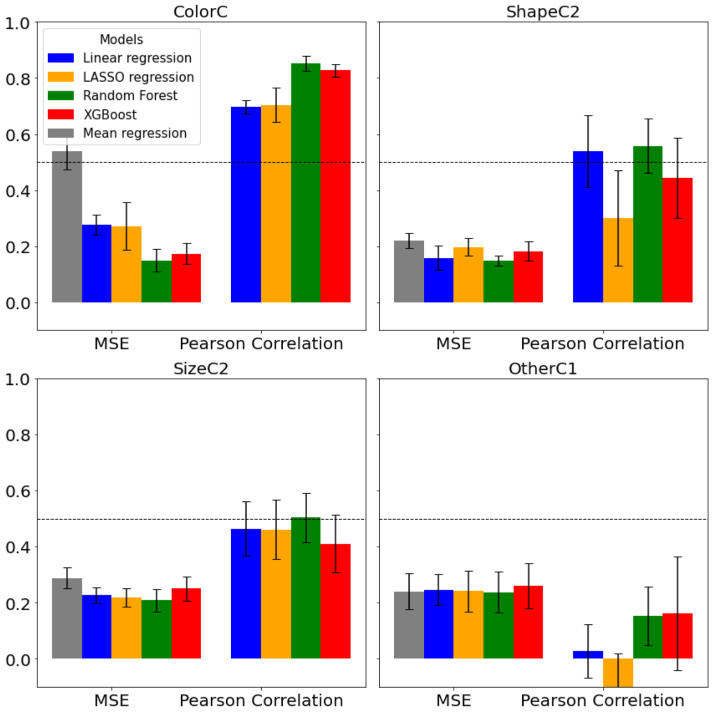


Figure 8.25: Comparison of model performance after tuning

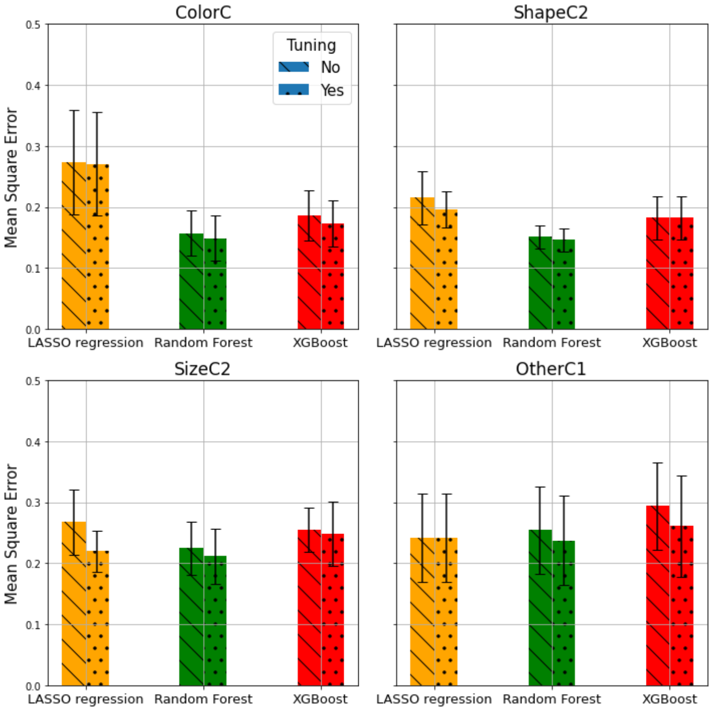


Figure 8.26: Comparison of MSE on test set before and after tuning

Model	Linear Regression	LASSO regression	Random Forest	XGBoost
Fitting Problem	Underfitting	Severe underfitting	Slight overfitting	Severe overfitting
Comment	1. Unable to distinguish clusters in color traits 2. Large errors on training sets		1. Overfitting was not due to insufficient training samples. 2. Tuning more hyperparameters did not avoid overfitting.	

Table 8.1: Summary of model performance

8.5. Prediction by Color

So far we have put all cultivars together to train the model and make predictions. Figure 8.4 indicated that cultivars of the same color clustered together on color traits. Thus, we categorized the cultivars into groups according to color to evaluate model performance across various color groups.

We started with the relationship between *ColorC* and *ColorD3* due to color clustering of cultivars on *ColorD3* and the correlation between *ColorC* and *ColorD3* in Figure 8.4. The *ColorD3* value was divided into intervals 10-30, 30-52, and 52-70 for red, orange, and yellow.

Fruit Color	$\hat{\beta}_1$	P-value	95% CI
All	-0.027	0	[-0.030, -0.023]
Red	0.178	0	[0.152, 0.204]
Orange	0.017	0.111	[-0.004, 0.038]
Yellow	0.018	0.206	[-0.010, 0.047]

Table 8.2: The $\hat{\beta}_1$ and its significance level for different colors.

Fitting linear regression model *ColorC* = $\beta_0 + \beta_1$*ColorD3* on different color groups, we derived four $\hat{\beta}_1$s and significance levels in Table 8.2. The overall regression indicated a significant negative correlation between *ColorC* and *ColorD3*: a unit increase in *ColorD3* leaded to a 0.027 unit decrease in *ColorC*. Pairwise regression showed *ColorC* and *ColorD3* significantly related only in red cultivars, with *ColorC* increasing by 0.178 units per unit rose in *ColorD3*. For yellow and orange cultivars, the correlations was weakly positive. Figure 8.27 illustrated the regression lines. The overall regression demonstrated Simpson's paradox as discussed in Section 2.3, highlighting cultivar color as a key predictor of traits.

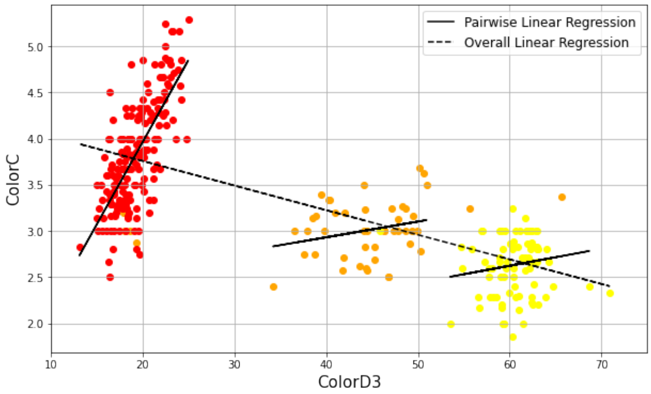


Figure 8.27: Overall and pairwise linear regressions of *ColorC* = $\beta_0 + \beta_1$*ColorD3*

The dataset was split into three color subsets for model training and evaluation. Figure 8.28 displayed the models' MSE on each set and the error bar was the standard deviation of MSE. The performance on the entire dataset served as a baseline. For *ColorC*, mean, linear, and LASSO regressions performed better on color subsets than the full dataset, showing they failed to capture cultivar color from $\mathbf{T}_{digital,ColorC}$. Fruit color helped these weak linear models. The random forest and XGBoost outperformed the baseline on orange and yellow subsets but underperformed on the red subset. Predicting *ColorC* for orange cultivars was easiest, while red cultivars need further investigation.

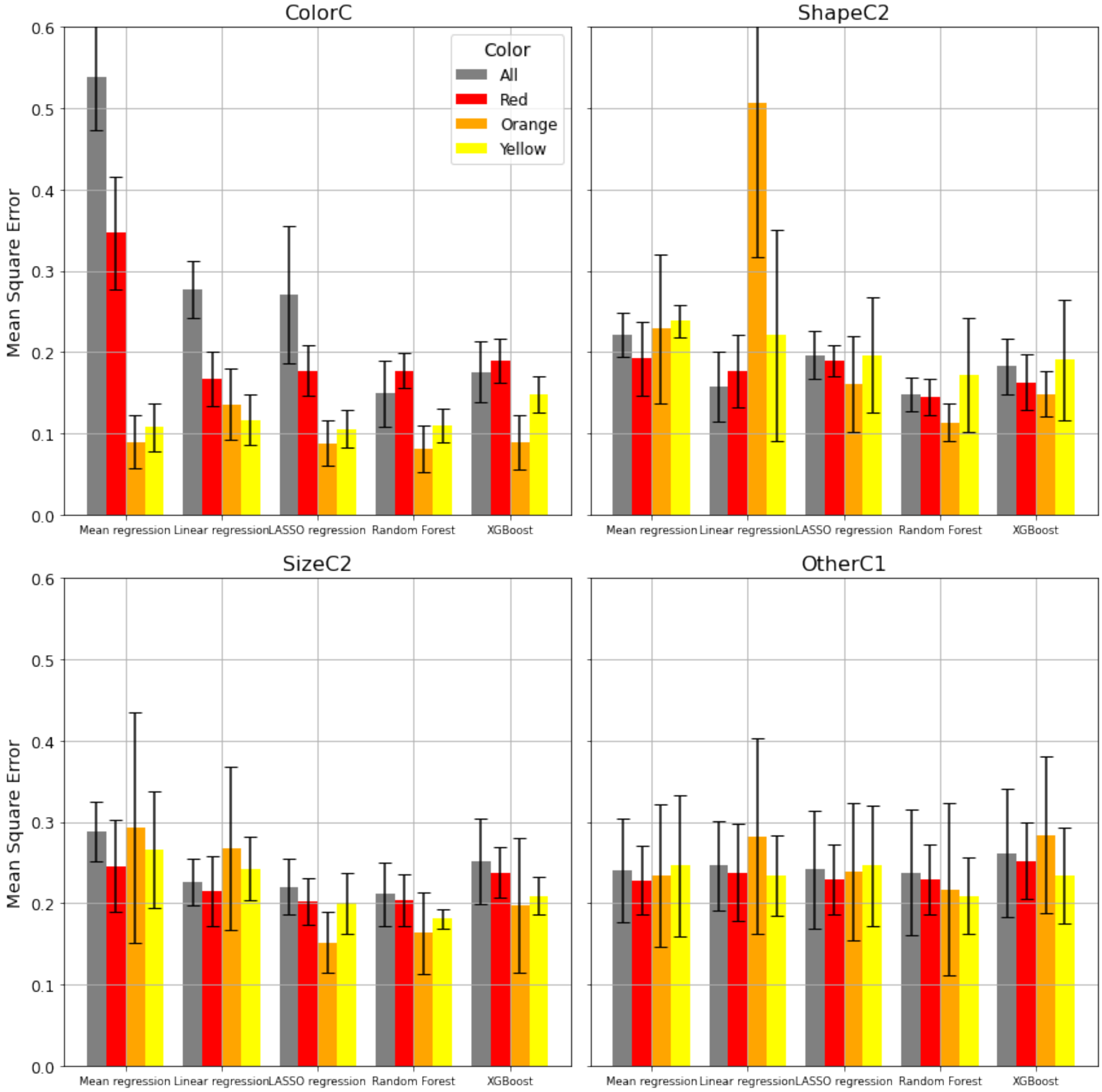


Figure 8.28: Performance of models on different color subsets

Predictive models performed differently on the other three traits. The linear regression of *ShapeC2* performed worse on color subsets than the baseline. However, LASSO regression performed better in subsets. For *SizeC2*, LASSO regression, random forest, and XGBoost performed slightly better in subsets.

Similarly, random forest also performed better than baseline in predicting *OtherC1* on subsets. In other cases, the baseline outperformed the worst subset but underperformed the best subset, indicating that fruit color could help the linear models (mean, linear, and LASSO regressions) to predict the color trait.

8.6. Low Quality Data from Tomatoes

The tomato also has both conventional and digital traits. Tomato's yield traits included conventional trait *C* and digital trait *D*, differing only in measurement methods. Each cultivar was measured once for *C* and twice for *D*, with two *D* measurements labeled as *D1* and *D2*. *C* and $\frac{1}{2}(D1+D2)$ were expected to have a high $\hat{\rho}^{Gen}$ due to representing the same phenotype measured differently. If $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}} \geq 0.8$, $\mathbf{t}_{con}$ could be replaced with $\mathbf{t}_{digital}$, so $\hat{\rho}^{Pearson}_{C,\frac{1}{2}(D1+D2)}$ should surpass 0.8.

The bi-trait model calculated $\hat{\rho}^{Gen}$s, with prior distributions and burn-in iterations as in Section 7.1. Table 8.3 shows $\hat{\rho}^{Gen}_{C,D}$ and $\hat{\rho}^{Pearson}_{C,D}$ with their 95% credible intervals.

$\mathbf{t}_{con}$	$\mathbf{t}_{digital}$	$\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ and 95% CI	$\hat{\rho}^{Pearson}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ and 95% CI
C	*D1*	0.77 (0.709, 0.846)	0.41 (0.354, 0.471)
	D2	0.65 (0.527, 0.772)	0.29 (0.153, 0.412)
	$\frac{1}{2}(D1+D2)$	0.72 (0.627, 0.805)	0.20 (0.136, 0.257)

Table 8.3: $\hat{\rho}^{Gen}$s and $\hat{\rho}^{Pearson}$s between conventional yield trait and digital yield traits

$\hat{\rho}^{Gen}$s from Table 8.3 were all below 0.8, and $\hat{\rho}^{Pearson}_{C,\frac{1}{2}(D1+D2)}$ was surprisingly 0.2. This was suspicious as *C* and $\frac{1}{2}(D1+D2)$ should be sampled from the same distribution. We used the Kolmogorov-Smirnov test to verify it.

We introduce the two-sample Kolmogorov-Smirnov test briefly. Given independent samples $X_1, X_2, \ldots, X_n$ from F and $Y_1, Y_2, \ldots, Y_m$ from G, the empirical cumulative distribution functions (ECDFs) are:

$$\widehat{F_n}(x) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \leq x) \text{ and } \widehat{G_m}(x) = \frac{1}{m}\sum_{j=1}^{m} I(Y_j \leq x).$$

where I is the indicator function that is equal to 1 if the condition is true and 0 otherwise. The test statistic is defined as: $\sup_x |\widehat{F_n}(x) - \widehat{G_m}(x)|$. It denotes the supremum of the absolute differences between the two ECDFs [58].

$$\mathcal{H}_0 : \text{Two samples are from the same distribution, } F = G.$$
$$\text{against } \mathcal{H}_1 : \text{Two samples are not from the same distributions, } F \neq G.$$

In our case, two independent samples were conventional trait values $C1_1, C1_2, \ldots C1_{N_{cultivars}}$ and the average digital trait values $\frac{1}{2}(D1_1 + D2_1), \frac{1}{2}(D1_2 + D2_2), \ldots, \frac{1}{2}(D1_{N_{cultivars}} + D2_{N_{cultivars}})$. The P-value was 1.13×10^{-43}, so we rejected the null hypothesis and concluded that two samples were not from the same distribution.

We were concerned about data reliability and reported the anomaly to the breeder, who agreed that $\hat{\rho}^{Gen}$s and $\hat{\rho}^{Pearson}$s were suspicious. Through stakeholder consultation and the analysis of ScaleCam pictures, it was discovered that an operational mistake resulted in immature tomatoes appearing in the pictures, which lead to inaccurate values of *D1* and *D2*. Conventional trait *C* values were accurate as they didn't rely on ScaleCam pictures.

9

Conclusion

This chapter will conclude the research by summarizing the key findings regarding the relationship between conventional and digital traits, along with their value and contributions to the application of digital phenotypes in plant breeding. In addition, it will address the limitations of the research and suggest opportunities for future researches.

This research completed the workflow for the relationship between digital and conventional plant traits and applied it to blocky peppers. We found some conventional traits could be replaced by single digital traits due to strong genetic correlations ($\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}} > 0.8$). For other conventional traits, some of them can be predicted using multiple digital traits through statistical learning models. It focused on four conventional traits and suggest that *ColorC*, *ShapeC2*, and *SizeC2* can be replaced or predicted by digital traits, reducing the need for future data collection. Only *OtherC1* still need to be collected in the future.

This research used the bi-trait model for $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$, which is a mixed effects model. Blocky peppers were classified into three different colors: red, yellow, and orange. Therefore, *red*, *yellow*, and *orange* were candidate variables for the fixed effect. We used the likelihood ratio test to compare the full model ($\mathbf{Y} = \mu + \mathbf{X}\beta + \mathbf{Z}u$) and the nested model ($\mathbf{Y} = \mu + \mathbf{Z}u$). The full model might be preferred when at least one of the traits was about color, while the nested model always outperformed if $\mathbf{t}_{con}$ and $\mathbf{t}_{digital}$ are both non-color traits. $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$ and $\hat{\rho}^{Pearson}_{\mathbf{t}_{con},\mathbf{t}_{digital}}$, two different correlations between $\mathbf{t}_{con}$ and $\mathbf{t}_{digital}$, were very different when both traits were about color.

Statistical learning models predicted conventional traits using digital traits. For each conventional trait $\mathbf{t}_{con}$, there were two predictor sets: $\mathbf{T}_{digital}$ and $\mathbf{T}_{digital,\mathbf{t}_{con}}$. $\mathbf{T}_{digital}$ contained all digital traits, while set $\mathbf{T}_{digital,\mathbf{t}_{con}}$ included digital traits meeting $\hat{\rho}^{Gen}_{\mathbf{t}_{con},\mathbf{t}_{digital}} > 0.2$. Models using $\mathbf{T}_{digital,\mathbf{t}_{con}}$ achieved similar accuracy to those using $\mathbf{T}_{digital}$ in MSE and $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$. As $\mathbf{T}_{digital,\mathbf{t}_{con}}$ had fewer predictors, it was preferred for prediction.

The research compared the performance of different models in predicting conventional traits. Linear and LASSO regression models were underfitted, especially in distinguishing between color traits, and experienced Simpson's paradox. On the other hand, random forest and XGBoost models tended to overfit, with XGBoost showing a more pronounced tendency. Even with efforts to tune more hyperparameters and increase the size of the training data, the overfitting problem remained unresolved. Despite overfitting, random forest and XGBoost performed better in test sets than underfitting models. Overall, the random forest model had the best performance due to its lower bias compared to linear and LASSO regressions and smaller variance than XGBoost.

This research revealed the possibility to revise the model performance threshold. By comparing MSE and $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}}$ of the statistical learning models with mean regression, it seems reasonable to raise the threshold from $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.5$ to $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.7$. This decision needs more analysis and consultation with breeders. Due to time constraints, we did not investigate this topic. Thus, $\hat{\rho}^{Pearson}_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.5$ remains the threshold in this research.

This research also predicted traits for color subsets separately. This method helped simple linear models (linear regression and LASSO regression) in predicting color traits but was ineffective for other traits or complex models. Among color subsets, traits of the orange subset are the easiest to predict, whereas more investigation is needed for the red ones.

There are two potential limitations to this research. The first one is model fitting issues. Linear regression and LASSO regression were underfitted while random forest and XGBoost were overfitted. Although the random forest was overfitted, its performance on the test set reached the desired level, so we did not further explore the fitting issues. This indicates potential risks in other situations. Secondly, the research failed to handle a key trait, *OtherC1* of blocky peppers. Models were unable to predict *OtherC1* even in the training set. Current digital traits are insufficient to predict *OtherC1*; more digital traits are needed for *OtherC1*.

It suggests that future research should explore predictive models of varying complexity, since tree-based models tended to overfit and linear models were underfitted on *ColorC*, *ShapeC2*, and *SizeC2*. Potential models are piecewise linear regression and polynomial regression, which lie between linear models and tree-based models in complexity. Besides, the reasonable model performance threshold is still a question to be answered.

Overall, this research explored the connection between digital and conventional phenotypes for plant breeding. Breeders have relied on conventional phenotypes for centuries, and the digital trait is a new tool for them. The digital phenotypes would help breeders to select superior varieties with understanding the relationship between two types of phenotypes.

Bibliography

[1] H. Li and J. Wang, "Biometrical approaches for analysis of phenotypic data of complex traits," *Phenomics in Crop Plants: Trends, Options and Limitations*, pp. 249–272, 2015.

[2] J. Kumar, A. Pratap, S. Kumar, *et al.*, *Phenomics in crop plants: Trends, options and limitations*. Springer, 2015.

[3] R. Pieruschka and U. Schurr, "Plant phenotyping: Past, present, and future," *Plant Phenomics*, vol. 2019, 2019. DOI: 10.34133/2019/7507131.

[4] E. Yol, C. Toker, and B. Uzun, "Traits for phenotyping," *Phenomics in crop plants: trends, options and limitations*, pp. 11–26, 2015.

[5] Q. Xiao, X. Bai, C. Zhang, and Y. He, "Advanced high-throughput plant phenotyping techniques for genome-wide association studies: A review," *Journal of advanced research*, vol. 35, pp. 215–230, 2022.

[6] R. S. Reshma and D. Das, "Molecular markers and its application in animal breeding," in *Advances in Animal Genomics*, Elsevier, 2021, pp. 123–140.

[7] L. E. Kruuk, J. Slate, J. M. Pemberton, S. Brotherstone, F. Guinness, and T. Clutton-Brock, "Antler size in red deer: Heritability and selection but no evolution," *Evolution*, vol. 56, no. 8, pp. 1683–1695, 2002.

[8] W. H. Finch, J. E. Bolin, and K. Kelley, *Multilevel modeling using R*. Chapman and Hall/CRC, 2019.

[9] A. van der Vaart, M. Jonker, and F. Bijma, *An introduction to mathematical statistics*. Amsterdam University Press, 2017.

[10] S. J. Kays, *Cultivated vegetables of the world: a multilingual onomasticon*. Springer, 2011.

[11] N. M. Laird and J. H. Ware, "Random-effects models for longitudinal data," *Biometrics*, pp. 963–974, 1982.

[12] C. R. Henderson, O. Kempthorne, S. R. Searle, and C. Von Krosigk, "The estimation of environmental and genetic trends from records subject to culling," *Biometrics*, vol. 15, no. 2, pp. 192–218, 1959.

[13] C. Gondro, J. Van der Werf, and B. Hayes, *Genome-wide association studies and genomic prediction*. Springer, 2013, vol. 1019.

[14] G. J. M. Rosa, 2019. [Online]. Available: https://si.biostat.washington.edu/sites/default/files/modules/Seattle-SISG-19-MM-Lecture04.pdf.

[15] G. de Los Campos, J. M. Hickey, R. Pong-Wong, H. D. Daetwyler, and M. P. Calus, "Whole-genome regression and prediction methods applied to plant and animal breeding," *Genetics*, vol. 193, no. 2, pp. 327–345, 2013.

[16] J. van der Werf, *2. principles of estimation of breeding values*, `https://jvanderw.une.edu.au/Chapter02_GENE422_EBV.pdf`, Accessed: 2024-9-23.

[17] D Byers, "Components of phenotypic variance," *Nature education*, vol. 1, no. 1, p. 161, 2008.

[18] A. Putz, 2018. [Online]. Available: `https://rpubs.com/amputz/Amatrix`.

[19] D. Lourenco, A. Legarra, and I. Aguilar, 2019. [Online]. Available: `https://nce.ads.uga.edu/wiki/lib/exe/fetch.php?media=ssgblup_day4_se.pdf`.

[20] K. Oldenbroek and L. van der Waaij, "Textbook animal breeding: Animal breeding and genetics for bsc students," 2014.

[21] R. Mrode, *Linear models for the prediction of animal breeding values*. Cabi, 2014.

[22] S. Forni, I. Aguilar, and I. Misztal, "Different genomic relationship matrices for single-step analysis using phenotypic, pedigree and genomic information," *Genetics Selection Evolution*, vol. 43, pp. 1–7, 2011.

[23] J Chen, *Use of mendelian sampling terms in genomic models*, 2009.

[24] J. D. Platten, J. N. Cobb, and R. E. Zantua, "Criteria for evaluating molecular markers: Comprehensive quality metrics to improve marker-assisted selection," *PloS one*, vol. 14, no. 1, e0210529, 2019.

[25] P. M. VanRaden, "Efficient methods to compute genomic predictions," *Journal of dairy science*, vol. 91, no. 11, pp. 4414–4423, 2008.

[26] M. Gadji *et al.*, "Nuclear remodeling as a mechanism for genomic instability in cancer," *Advances in cancer research*, vol. 112, pp. 77–126, 2011.

[27] A. Legarra, I. Aguilar, and I. Misztal, "A relationship matrix including full pedigree and genomic information," *Journal of Dairy Science*, vol. 92, no. 9, pp. 4656–4663, 2009, ISSN: 0022-0302. DOI: `https://doi.org/10.3168/jds.2009-2061`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0022030209707933`.

[28] A. Legarra, O. F. Christensen, I. Aguilar, and I. Misztal, "Single step, a general approach for genomic selection," *Livestock Science*, vol. 166, pp. 54–65, 2014.

[29] O. F. Christensen and M. S. Lund, "Genomic prediction when some animals are not genotyped," *Genetics Selection Evolution*, vol. 42, pp. 1–8, 2010.

[30] I Aguilar, I Misztal, D. Johnson, A. Legarra, S Tsuruta, and T. Lawlor, "Hot topic: A unified approach to utilize phenotypic, full pedigree, and genomic information for genetic evaluation of holstein final score," *Journal of dairy science*, vol. 93, no. 2, pp. 743–752, 2010.

[31] H. B. Zaabza, M. Taskinen, E. A. Mäntysaari, T. Pitkänen, G. P. Aamand, and I. Strandén, "Breeding value reliabilities for multiple-trait single-step genomic best linear unbiased predictor," *Journal of Dairy Science*, vol. 105, no. 6, pp. 5221–5237, 2022.

[32] H. Önder *et al.*, "Multi-trait single-step genomic prediction for milk yield and milk components for polish holstein population," *Animals*, vol. 13, no. 19, p. 3070, 2023.

[33] H. Ben Zaabza, M. Taskinen, E. A. Mäntysaari, T. Pitkänen, G. P. Aamand, and I. Strandén, "Breeding value reliabilities for multiple-trait single-step genomic best linear unbiased predictor," *Journal of Dairy Science*, vol. 105, no. 6, pp. 5221–5237, 2022, ISSN: 0022-0302. DOI: `https://doi.org/10.3168/jds.2021-21016`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S0022030222002119`.

[34] M. Weigt and H. Szurmant, "Genetic covariance," in *Brenner's Encyclopedia of Genetics (Second Edition)*, S. Maloy and K. Hughes, Eds., Second Edition, San Diego: Academic Press, 2013, pp. 242–245, ISBN: 978-0-08-096156-9. DOI: `https://doi.org/10.1016/B978-0-12-374984-0.00613-6`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/B9780123749840006136`.

[35] M. Lynch, B. Walsh, *et al.*, *Genetics and analysis of quantitative traits*. Sinauer Sunderland, MA, 1998, vol. 1.

[36] A. Zellner, *An introduction to bavesian inference in econometrics. new york: John wilev & sons*, 1971.

[37] P. Pérez-Rodríguez and G. de Los Campos, "Multitrait bayesian shrinkage and variable selection models with the bglr-r package," *Genetics*, vol. 222, no. 1, iyac112, 2022.

[38] Z. Zhang, "A note on wishart and inverse wishart priors for covariance matrix," *Journal of Behavioral Data Science*, vol. 1, no. 2, pp. 119–126, 2021.

[39] K. P. Murphy, "Conjugate bayesian analysis of the gaussian distribution," *def*, vol. 1, no. $2\sigma 2$, p. 16, 2007.

[40] S. Hug, M. Schwarzfischer, J. Hasenauer, C. Marr, and F. J. Theis, "An adaptive scheduling scheme for calculating bayes factors with thermodynamic integration using simpson's rule," *Statistics and Computing*, vol. 26, pp. 663–677, 2016.

[41] C. W. Fox and S. J. Roberts, "A tutorial on variational bayesian inference," *Artificial intelligence review*, vol. 38, pp. 85–95, 2012.

[42] A. E. Raftery and S. M. Lewis, "Implementing mcmc," *Markov chain Monte Carlo in practice*, pp. 115–130, 1996.

[43] C. Andrieu and J. Thoms, "A tutorial on adaptive mcmc," *Statistics and computing*, vol. 18, pp. 343–373, 2008.

[44] D. Sorensen and D. Gianola, *Likelihood, Bayesian, and MCMC methods in quantitative genetics*. Springer Science & Business Media, 2007.

[45] S. Geman and D. Geman, “Stochastic relaxation, gibbs distributions, and the bayesian restoration of images,” *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 721–741, 1984.

[46] G. Casella and E. I. George, “Explaining the gibbs sampler,” *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[47] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 58, no. 1, pp. 267–288, 1996.

[48] J. Gareth, W. Daniela, H. Trevor, and T. Robert, *An introduction to statistical learning: with applications in R*. Spinger, 2013.

[49] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer, 2009, vol. 2.

[50] A. Cutler, D. R. Cutler, and J. R. Stevens, “Random forests,” *Ensemble machine learning: Methods and applications*, pp. 157–175, 2012.

[51] L. Breiman, “Some infinity theory for predictor ensembles,” Citeseer, Tech. Rep., 2000.

[52] T. K. Ho, “Random decision forests,” in *Proceedings of 3rd international conference on document analysis and recognition*, IEEE, vol. 1, 1995, pp. 278–282.

[53] H. Ishwaran, “The effect of splitting on random forests,” *Machine learning*, vol. 99, pp. 75–118, 2015.

[54] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 2016, pp. 785–794.

[55] A. K. R. Choudhury, *Principles of colour and appearance measurement: Visual measurement of colour, colour comparison and management*. Woodhead Publishing, 2014.

[56] G. Biau and E. Scornet, “A random forest guided tour,” *Test*, vol. 25, pp. 197–227, 2016.

[57] L. E. Yelle, “The learning curve: Historical review and comprehensive survey,” *Decision sciences*, vol. 10, no. 2, pp. 302–328, 1979.

[58] V. W. Berger and Y. Zhou, “Kolmogorov–smirnov test: Overview,” *Wiley statsref: Statistics reference online*, 2014.

[59] D. F. Andrews and C. L. Mallows, “Scale mixtures of normal distributions,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 1, pp. 99–102, 1974.

[60] E. I. George and R. E. McCulloch, “Variable selection via gibbs sampling,” *Journal of the American Statistical Association*, vol. 88, no. 423, pp. 881–889, 1993.

[61] F. Zou, H. Huang, S. Lee, and I. Hoeschele, “Nonparametric bayesian variable selection with applications to multiple quantitative trait loci mapping with epistasis and gene–environment interaction,” *Genetics*, vol. 186, no. 1, pp. 385–394, 2010.

[62] T. H. Meuwissen, B. J. Hayes, and M. Goddard, “Prediction of total genetic value using genome-wide dense marker maps,” *genetics*, vol. 157, no. 4, pp. 1819–1829, 2001.

[63] D. Habier, R. L. Fernando, K. Kizilkaya, and D. J. Garrick, “Extension of the bayesian alphabet for genomic selection,” *BMC bioinformatics*, vol. 12, pp. 1–12, 2011.

[64] M. John and A. Mieldzioc, “The comparison of the estimators of banded toeplitz covariance structure under the high-dimensional multivariate model,” *Communications in Statistics-Simulation and Computation*, vol. 49, no. 3, pp. 734–752, 2020.

[65] M. Janiszewska, “Structures of the covariance matrix: An overview,” *Biometrical Letters*, vol. 59, no. 2, pp. 141–157, 2022.

A

Estimated PDF of Digital Trait Values and Statistical Summary of All Traits

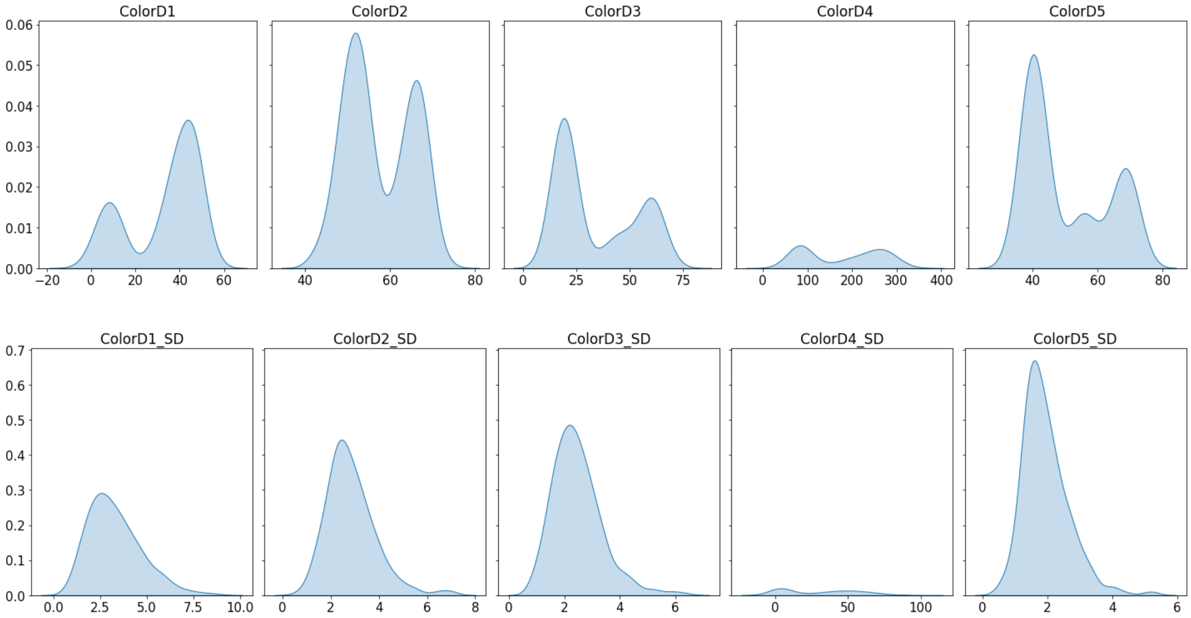


Figure A.1: Estimated distributions of digital color traits

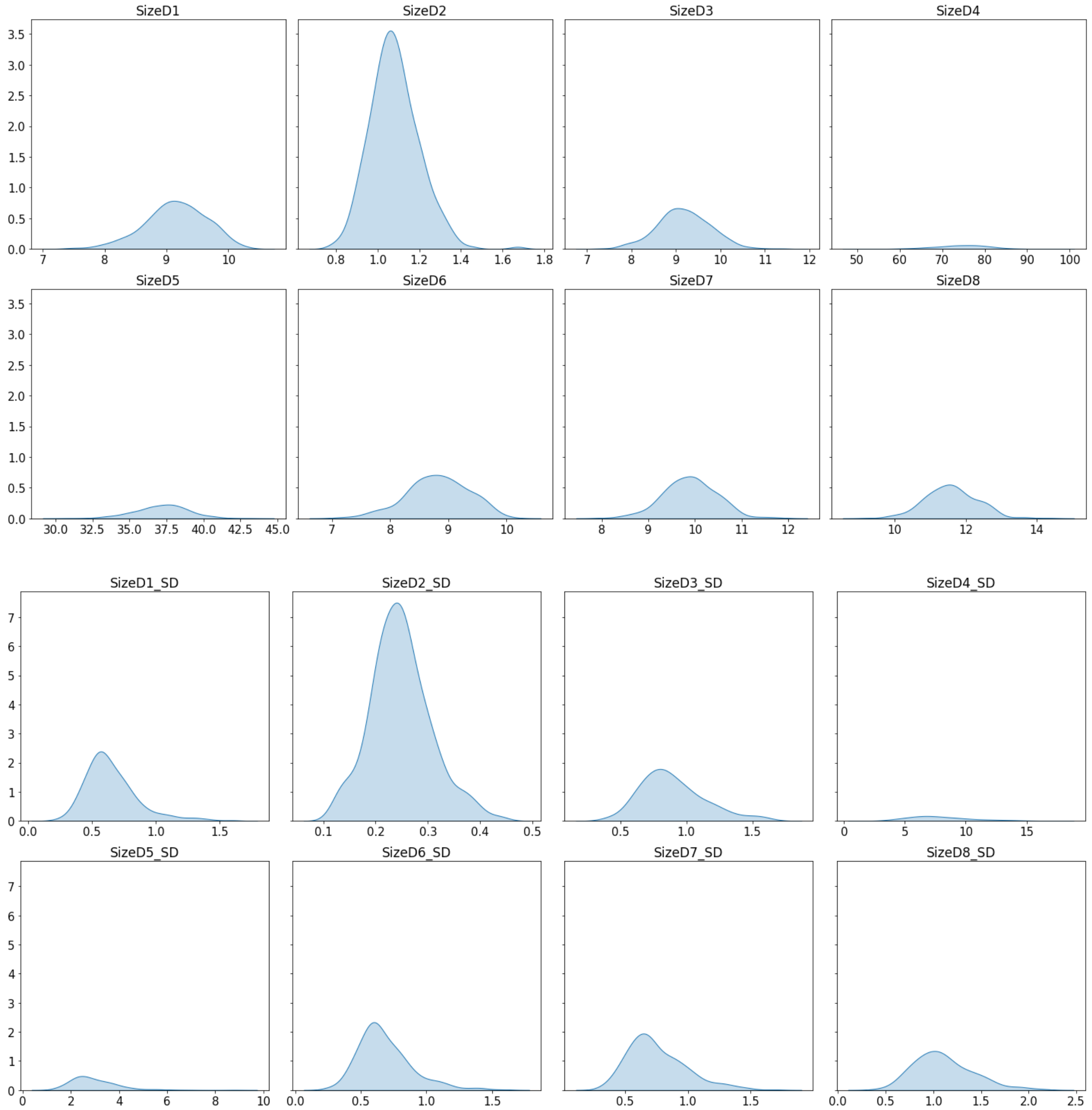


Figure A.2: Estimated distributions of digital size traits

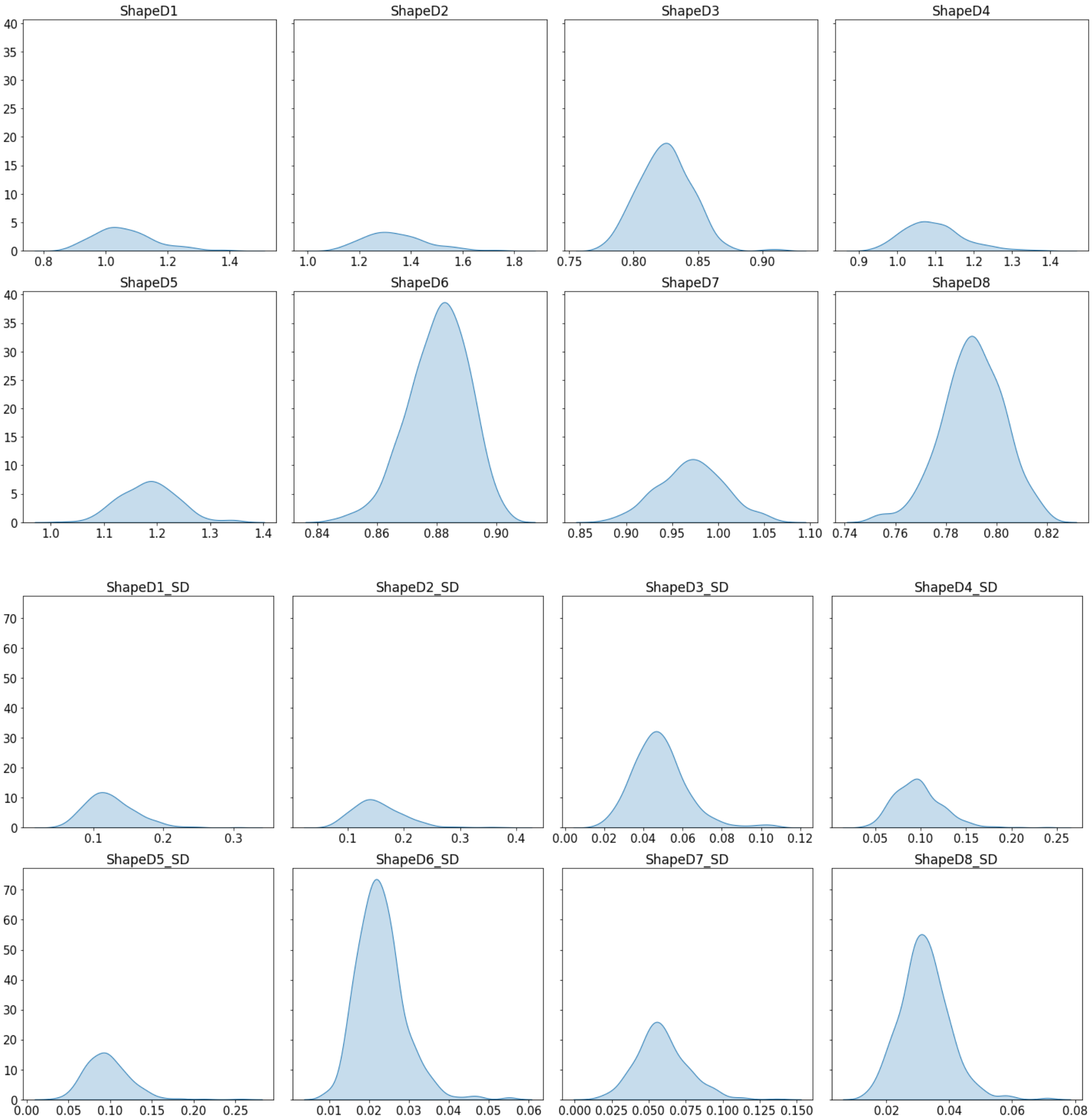


Figure A.3: Estimated distributions of digital shape traits

Trait	Mean	Standard Deviation	Min	25%	50%	75%	Max
ColorC	3.39	0.73	1.86	2.83	3.29	3.88	5.29
ShapeC1	3.74	0.49	2.75	3.40	3.70	4.05	5.25
ShapeC2	4.46	0.47	2.80	4.14	4.44	4.75	5.71
SizeC1	2.92	0.57	1.71	2.50	2.86	3.25	n 4.89
SizeC2	4.80	0.53	3.20	4.45	4.86	5.17	6.25
OtherC1	3.06	0.49	1.00	3.00	3.00	3.33	5.00
OtherC2	1.11	0.40	1.00	1.00	1.00	1.00	4.60
ColorD1	32.78	16.14	-3.70	13.38	38.51	45.62	55.08
ColorD2	57.41	7.70	41.70	51.11	54.79	65.19	73.70
ColorD3	34.00	18.57	13.09	18.38	22.12	56.60	70.87
ColorD4	176.34	85.84	42.89	85.15	190.35	260.51	325.13
ColorD5	50.12	12.52	35.06	39.77	42.67	65.52	72.31
SizeD1	9.14	0.50	7.47	8.85	9.16	9.48	10.27
SizeD2	1.08	0.12	0.78	1.00	1.07	1.15	1.67
SizeD3	9.17	0.60	7.32	8.78	9.16	9.56	11.39
SizeD4	74.41	6.47	52.80	70.43	74.92	78.82	94.91
SizeD5	37.16	1.81	30.84	36.13	37.36	38.32	43.07
SizeD6	8.81	0.53	7.12	8.46	8.82	9.18	10.08
SizeD7	9.89	0.58	8.00	9.51	9.90	10.26	11.86
SizeD8	11.59	0.74	9.24	11.08	11.57	12.04	14.35
ShapeD1	1.06	0.10	0.87	0.99	1.05	1.12	1.42
ShapeD2	1.34	0.12	1.11	1.25	1.33	1.41	1.76
ShapeD3	0.82	0.02	0.77	0.81	0.82	0.84	0.91
ShapeD4	1.10	0.08	0.94	1.04	1.08	1.13	1.39
ShapeD5	1.18	0.05	1.02	1.15	1.19	1.22	1.35
ShapeD6	0.88	0.01	0.85	0.87	0.88	0.89	0.90
ShapeD7	0.97	0.04	0.88	0.95	0.97	0.99	1.06
ShapeD8	0.79	0.01	0.75	0.78	0.79	0.80	0.82
ColorD1_SD	3.29	1.43	0.69	2.24	3.02	4.12	8.78
ColorD2_SD	2.84	1.02	0.64	2.16	2.69	3.35	7.06
ColorD3_SD	2.46	0.90	0.82	1.85	2.31	2.92	6.37
ColorD4_SD	30.84	25.42	0.67	3.67	32.89	52.08	91.09
ColorD5_SD	1.94	0.70	0.43	1.46	1.79	2.27	5.29
SizeD1_SD	0.65	0.21	0.23	0.52	0.61	0.75	1.60
SizeD2_SD	0.25	0.06	0.12	0.21	0.24	0.28	0.44
SizeD3_SD	0.90	0.24	0.39	0.73	0.86	1.03	1.64
SizeD4_SD	7.92	2.61	2.80	6.08	7.38	9.35	16.41
SizeD5_SD	2.95	0.99	1.30	2.27	2.73	3.48	8.81
SizeD6_SD	0.68	0.21	0.26	0.55	0.64	0.78	1.59
SizeD7_SD	0.76	0.23	0.33	0.60	0.70	0.89	1.68
SizeD8_SD	1.11	0.31	0.41	0.88	1.07	1.30	2.17
ShapeD1_SD	0.12	0.04	0.05	0.10	0.12	0.15	0.31
ShapeD2_SD	0.16	0.05	0.07	0.12	0.15	0.18	0.38
ShapeD3_SD	0.05	0.01	0.02	0.04	0.05	0.06	0.11
ShapeD4_SD	0.10	0.03	0.04	0.08	0.10	0.11	0.24
ShapeD5_SD	0.10	0.03	0.04	0.08	0.09	0.11	0.26
ShapeD6_SD	0.02	0.01	0.01	0.02	0.02	0.03	0.06
ShapeD7_SD	0.06	0.02	0.02	0.05	0.06	0.07	0.14
ShapeD8_SD	0.03	0.01	0.01	0.03	0.03	0.04	0.07

Table A.1: Statistical summary of trait values

B

Conditional Distributions of A Multivariate Normal Distribution

In this section, we follow the work given by Sorensen and Gianola [44, page 42]. It is assumed $X \sim \mathcal{N}(\mu, \Sigma)$ is a multivariate normal vector. Consider partitioning X, μ and Σ into

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}.$$

Then, the conditional distribution of the first partition given the second is

$$p(X_1 \mid X_2 = x_2) = \mathcal{N}(\mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}).$$

Proof. Define $Z = X_1 + AX_2$ where $A = -\Sigma_{12}\Sigma_{22}^{-1}$. Firstly, we can show Z and X_2 are uncorrelated:

$$\begin{aligned} \mathrm{Cov}(Z, X_2) &= \mathrm{Cov}(X_1, X_2) + \mathrm{Cov}(AX_2, X_2) \\ &= \Sigma_{12} + A\mathrm{Var}(X_2) \\ &= \Sigma_{12} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22} \\ &= 0. \end{aligned}$$

Z and X_2 have jointly normal distribution and are uncorrelated. Therefore Z and X_2 are independent.

The conditional expectation of $(X_1 \mid X_2 = x_2)$ is

$$\begin{aligned}
\mathbb{E}(X_1 \mid X_2 = x_2) &= \mathbb{E}(z - AX_2 \mid X_2 = x_2) \\
&= \mathbb{E}(z \mid X_2 = x_2) - \mathbb{E}(AX_2 \mid X_2 = x_2) \\
&= \mathbb{E}(z) - Ax_2 \\
&= \mu_1 + A\mu_2 - Ax_2 \\
&= \mu_1 + A(\mu_2 - x_2) \\
&= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2).
\end{aligned}$$

The conditional covariance matrix is

$$\begin{aligned}
\mathrm{Var}(X_1 \mid X_2 = x_2) &= \mathrm{Var}(Z - AX_2 \mid X_2 = x_2) \\
&= \mathrm{Var}(Z \mid X_2 = x_2) + \mathrm{Var}(AX_2 \mid X_2 = x_2) - A\mathrm{Cov}(Z, -X_2 \mid X_2 = x_2) - \mathrm{Cov}(Z, -X_2 \mid X_2 = x_2)A^T \\
&= \mathrm{Var}(Z \mid X_2 = x_2) \\
&= \mathrm{Var}(Z) \\
&= \mathrm{Var}(Z_1 + AX_2) \\
&= \mathrm{Var}(X_1) + A\mathrm{Var}(X_2)A^T + A\mathrm{Cov}X_1, X_2 + \mathrm{Cov}(X_2, X_1)A^T \\
&= \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{22}\Sigma_{22}^{-1}\Sigma_{21} - 2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\
&= \Sigma_{11} + \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} - 2\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} \\
&= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.
\end{aligned}$$

We used the property of covariance matrices:

$$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + \mathrm{Cov}(X, Y) + \mathrm{Cov}(Y, X).$$

□

C

Other Prior Distributions of Random Effects and Covariance

C.1. Priors of random effects

Gaussian priors (or Normal priors): The Gaussian distribution has two parameters: mean and variance (σ_z^2). Mean is set to zero because any nonzero mean for a term in the random effects is expressed as part of the fixed effect. Therefore, $\omega = \sigma_z^2$. With Gaussian priors, $\hat{u}$ is the BLUP of u [15]. The posterior distribution of random effects, $p(u \mid \mathbf{Y}, \mu, \sigma_\varepsilon^2, \sigma_z^2) \propto \prod_{i=1}^{N} \mathcal{N}(\mathbf{Y}_i \mid \mu + \sum_{j=1}^{q} \mathbf{Z}_{ij} u_j, \sigma_\varepsilon^2) \prod_{j=1}^{q} p(u_j \mid 0, \sigma_z^2)$ is multivariate normal, with posterior mean given by $\hat{u} = [\mathbf{Z}^T \mathbf{Z} + \sigma_z^2 \sigma_\varepsilon^2 \mathbf{I}]^{-1} \mathbf{Z}^T (\mathbf{Y} - \mu)$ which is the BLUP of random effect coefficients [15].

Heavy-tailed priors: Scaled t and double exponential are two commonly used thick-tailed priors. Compared with Gaussian, these distributions have higher mass at zero and thicker tails. Therefore they include strong shrinkage toward zero of estimates with small effects and less shrinkage of estimates with sizable effects. For computational convenience, the thick-tail densities are commonly represented as infinite mixtures of scale normal densities of the form $p(u_j \mid \omega) = \int \mathcal{N}(u_j \mid 0, \sigma_{z_j}^2) p(\sigma_{z_j}^2 \mid \omega) \partial \sigma_{z_j}^2$, where $p(\sigma_{z_j}^2 \mid \omega)$ is prior distribution assigned to random effects variance parameters [59]. The posterior marginal prior of u_j is scaled t distribution if $p(\sigma_{z_j}^2 \mid \omega)$ is a scaled inverse chi-square distribution. It is double exponential distribution if $p(\sigma_{z_j}^2 \mid \omega)$ is an exponential distribution. Double-exponential density has only one parameter: rate. While scaled t distribution is indexed by two parameters: scale and degree of freedom which gives the scaled t more flexibility to control tails thickness. By using priors that are finite mixtures, an even higher degree of flexibility of the shape of the prior can be obtained [15].

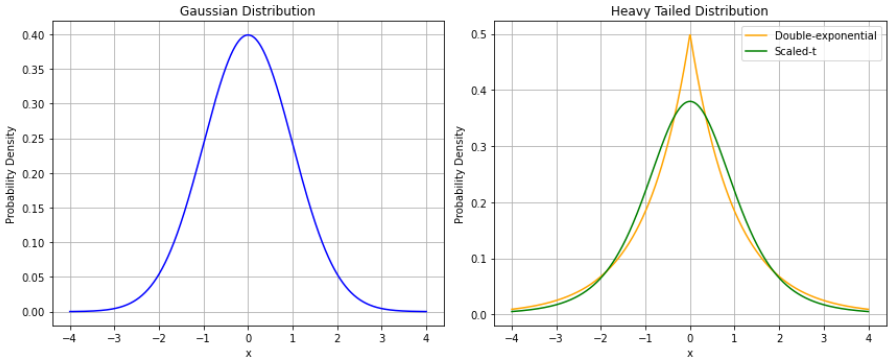


Figure C.1: Gaussian and heavy-tailed distributions

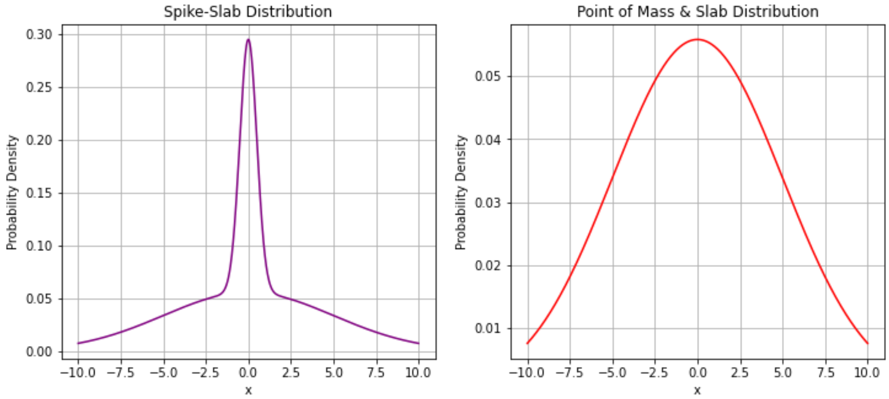


Figure C.2: Spike–slab and Point of Mass & Slab Distributions

Spike–slab priors: Spike–slab priors are mixtures of two densities: one with small variance (the spike) and one with large variance (the slab) [60]. The spike and the slab are both zero-mean densities. A general form of this prior is $\pi \times N\left(0, \sigma_1^2\right) + (1-\pi) \times N\left(0, \sigma_2^2\right)$ where $\pi \in [0,1]$ and σ_1^2 and σ_2^2 are variance parameters. Spike–slab priors are Gaussian priors when $\pi = 0$ or 1. Apart from mixing two Gaussian distributions, Spike–slab priors can be obtained by mixing other densities such as scaled t or double exponential [61].

Point of mass at zero and slab priors: Spike–slab priors are a point of mass at zero and a slab priors when σ_1^2 or $\sigma_2^2 \longrightarrow 0$, in this case, the small-variance component of the mixture collapses to a point of mass at zero. Point of mass at zero and slab priors are used to induce a combination of variable selection and shrinkage. These priors are used in model BayesB where the slab is a scaled-t density and model BayesC where the slab is a normal density [62, 63]. More information about BayesB and BayesC models is introduced by Habier et al. [63].

C.2. Priors of covariance parameters

Spherical structure: The covariance matrix $\mathbf{\Omega}$ is in spherical structure if it is proportional to the identity matrix. Spherical structure can be expressed as

$$\mathbf{\Omega} = \begin{pmatrix} \alpha & 0 & \cdots & 0 \\ 0 & \alpha & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha \end{pmatrix},$$

where α is a constant. A spherical covariance matrix implies that all components of the observation vector not only share the same variance but are also independent of each other. It is the smallest linear structure with one dimension of the space structure. Therefore, it simplifies the estimation problem to just one unknown parameter (α). The spherical covariance structure is frequently used as the target matrix within shrinkage methods for the simple structure [64, 65].

Diagonal structure: The covariance matrix $\mathbf{\Omega}$ with diagonal structure is defined as:

$$\mathbf{\Omega} = \begin{pmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \alpha_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \alpha_m \end{pmatrix}.$$

The diagonal structure provides more flexibility than the spherical structure. The components of the observation vector can have heterogenous variances and are independent. The dimension of the structure space is equal to m. Similar to spherical structure, diagonal structure of the covariance matrix is also used to as the target matrix in the shrinkage method [65].