THOMPSON RIVERS UNIVERSITY

Identification of Important SNPs using Penalized Models and Bayesian Deep Learning on Whole-Genome *Arabidopsis Thaliana* Data

By

Nikita Kohli

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Science in Data Science

Kamloops, British Columbia

July 2023

SUPERVISORS

Dr. Jabed Tomal

Dr. Yan Yan

ABSTRACT

The process of identifying the most important and informative features from a data set for a particular task is known as feature selection. Feature selection is a critical problem for statistical modeling and machine learning since employing all features might result in over-fitting. In high-dimensional data, where the number of features can be significantly greater than the number of samples, feature selection is even more difficult. One such application where the high-dimensional challenge is common is in Genome-Wide Association Studies (GWAS). GWAS aims to identify the relationship between genetic variations, such as Single Nucleotide Polymorphisms (SNPs), and physical traits. Feature selection algorithms based on statistical and machine learning methods are often used to tackle the high-dimensionality problem. This research aims to tackle this challenge by proposing two workflows to identify several potentially important SNPs. The first workflow, PentaPen, combines five penalized models - Ridge, LASSO, and Elastic net using all SNPs and Group LASSO and Sparse Group Lasso (SGL) using filtered SNPs (union of SNPs selected by Ridge, LASSO, and Elastic net). The second workflow, BayesDL, combines Bayesian methods with deep learning using preliminary filtered SNPs found by Chi-square and ANOVA as input. PentaPen, a machine learning model, aims to provide reduced numbers of SNPs by leveraging the beneficial properties of five penalized models. The union of SNPs selected by Group LASSO and SGL are the output SNPs of PentaPen. BayesDL, a cascaded deep learning model, along with identifying important SNPs, aims to provide high prediction performance. BayesDL also mitigates the issue of over-fitting while handling data sets with fewer sample sizes, a limitation in various traditional neural networks.

The performances of the proposed workflows are compared with the existing methodologies based on the quality metrics, Precision, Recall, F1 score, AUC, R-squared, RMSE, and Accuracy. The systematic comparison of single penalized models provides a guideline for researchers to make informed decisions to choose a penalized model. BayesDL's performance is compared with the Convolutional neural network (CNN). In addition, the important SNPs from both workflows are validated to locate genes; these are compared with the output SNPs or genes between each other and from GWAS software (GAPIT and TASSEL).

Findings of the continuous and categorical phenotype of *Arabidopsis thaliana* plant data indicate that PentaPen performs similarly to LASSO and Elastic Net while better than Ridge, Group LASSO, and SGL by reducing over-fitting. Reduced over-fitting was evident with a 10% decrease in the testing metrics compared to the training metrics. PentaPen performs similarly to Ridge, LASSO, and Elastic Net for the binary phenotype. BayesDL performs better than CNN for all the phenotypes. The findings from the proposed workflows complement with GWAS, using different models (generalized linear models in GAPIT and TASSEL versus penalized models and probabilistic models in two proposed workflows respectively).

My study provides a classifier and regressor - PentaPen - for researchers finding reduced numbers of important SNPs for further analysis; the study also provides a rigorous comparison of penalized models to gain insights into the strengths and predictive performance of each model. Furthermore, the study also gives the bioinformatics community a cascaded classifier and regressor, BayesDL or Bayesian Neural Network (BNN), useful for the prediction and identification of important SNPs in the whole-genome SNP data.

Key Words: Genomic Wide Association Study; Single Nucleotide Polymorphism; SNP Identification; Machine Learning; Deep Learning; High Dimensional Data.

ACKNOWLEDGEMENTS

I first express my gratitude to my supervisors, Dr. Jabed Tomal and Dr. Yan Yan, who guided me through this thesis. They trusted me to build on their collective knowledge, which serves as the basis of this research.

I thank Dr. Qinglin (Roger) Yu, the MSc Data Science program coordinator, for his assistance in keeping me on schedule. I would like to extend my profound gratitude to the faculty at Thompson Rivers University, particularly Dr. Emad Mohammed, Dr. Yue Zhang, Dr. Mateen Shaikh, Dr. Mila Kwiatkowska, Dr. Piper Jackson, Dr. Erfanul Hoque, and Dr. Trent Tucker for their patience and support.

I acknowledge the Program of Data Science at Thompson Rivers University and Compute Canada for hosting the 22GB and 32GB Linux servers respectively which are used for computation in this research. I would also like to acknowledge the funding for this research from:

1. Natural Sciences and Engineering Research Council of Canada (NSERC) awarded to Dr. Jabed Tomal, Department of Mathematics and Statistics, and Dr. Yan Yan, Department of Computing Science respectively, Thompson Rivers University.

Finally, I express my gratitude to the God, my parents, my sister, and my brother for their everlasting love and support.

Contents

List of Figures

List of Tables

Nomenclature

ABNN Approximate Bayesian Neural Network

ANN Artificial Neural Network

ANOVA Analysis of Variance

BLINK Bayesian information and Linkage disequilibrium iteratively nested keyway

BLUP Best Linear Unbiased Prediction

BNN Bayesian Neural Network

BRR Bayesian Ridge Regression

CAR Correlation-Adjusted marginal correlation

CNN Convolutional Neural Network

CoV Coefficient of Variation

CV Cross-validation

DL Deep Learning

DNA Deoxyribonucleic acid

DNN Deep Neural Network

FarmCPU Fixed and Random Model Circulating Probability Unification

GAPIT Genome Association and Prediction Integrated Tool

GBM Gradient Boosting Machine

GBT Gradient Boosting Tree

GLM Generalized Linear Model

GS Genomic Selection

GWAS Genome Wide Association Studies

HMC Hamiltonian Monte-Carlo

ICA Independent Component Analysis

LASSO Least Absolute Shrinkage and Selection Operator

LDA Linear Discriminant Analysis

LD Linkage Disequilibrium

LMM Linear Mixed Model

MA Metropolis Algorithm

MCMC Markov Chain Monte Carlo

MLMM Multiple Loci Mixed Model

ML Machine Learning

NEG Normal Exponential Gamma

NUTS No-U Turn

PCA Principal Component Analysis

PMR Penalized Multiple Regression

PRS Polygenic Risk Scores

PUMA Penalized Unified Multiple Analysis

RF Random Forest

RLS Regularized Least Squares

SGL Sparse Group LASSO

SNP Single Nucleotide Polymorphisms

SSVS Stochastic Search Variable Selection

SVM Support Vector Machine

t-SNE t-distributed Stochastic Neighbor Embedding

TASSEL Trait Analysis by aSSociation, Evolution and Linkage

UMAP uniform manifold approximation and projection (UMAP)

VCF Variant Call Format

VI Variational Inference

WES Whole-Exome Sequencing

WGS Whole-Genome Sequencing

Chapter 1

Introduction

Feature selection is a process of identifying and selecting the most informative and relevant features in a data set. It is an important issue in statistical modeling and machine learning since using all features might result in over-fitting and poor model performance. In high-dimensional data sets, this problem becomes particularly challenging, where the number of features or variables is much larger than the number of samples. Due to the high dimensionality, the data matrix is sparse and has a lot of noise or irrelevant features, which might lead to the models' poor performance. Genome-wide association studies (GWAS) is a popular bioinformatics application of a high-dimensional data set where feature selection is challenging. GWAS discovers associations between specific DNA variants (Single Nucleotide Polymorphism (SNP)) and phenotype. SNP is the key to understanding the genetic causes of an organism's physical traits (Allen et al. [2014]). The substitution of a C for a G in the nucleotide sequence AACGAT to produce the sequence AACCAT is an example of an SNP (Britannica [2019]).

There are a few limitations while performing association tests using whole-genome SNP data. On extensive complex data, GWAS methods frequently have constraints on the SNP scale and may fail to detect associated SNPs (Korte and Farlow [2013]). In addition, the sparsity of data in high-dimensional domains leads to the "curse of dimensionality" (Bellman and Kalaba [1959]). In other words, it occurs for whole-genome SNP data when the number of SNPs (p) is much larger than the number of samples (n) which could be one of the possible challenges for researchers to perform the analysis.

1.1 Motivation

Although several studies have been conducted in the past using machine learning and deep learning algorithms on human whole-genome SNP data, there is limited research focused on plant data. This limitation is noteworthy because plant genomes exhibit distinct genetic characteristics when compared to human genomes. Plant genomes are more complex and large in dimension; they have a higher degree of homozygosity. Moreover, the extended growth period of plants, such as wheat requiring 120 days, necessitates longer research timelines. The constrained research budget also plays a role, with researchers often prioritizing the study of animal species over plants (Xu et al. [2014]). As a result, plant genomics research faces challenges related to genetic complexity and computational time to perform classification and regression. These differences lead to looking for different statistical approaches and tuning parameters. This research aims to address the challenge to identify important SNPs from plant whole-genome SNP data. This study uses

the model organism, *Arabidopsis thaliana* due to its faster growth rate, high genetic diversity, fully sequenced genome, and well-characterized SNP data sets available. It is a small flowering plant that is a member of the mustard (Brassicaceae) family, which includes cultivated species such as cabbage and radish.

1.2 Scope

One possible way to solve the dimensionality problem and reduce the number of loci associated with the trait (increases the detection power) is to use feature selection. Supervised feature selection is typically used for classification or regression applications on whole-genome SNP data to find potentially important SNPs (or features). It attempts to choose a subset of features that can classify data into separate groups or measure up to the regression targets (Li et al. [2017]). A previous study from our group (Puliparambil et al. [2022]) investigated the curse of dimensionality issue on high-dimensional scRNA-seq data.

Wrapper methods (forward, backward, and step-wise selection), filter methods (Analysis of variance (ANOVA), Pearson correlation, variance thresholding), and embedded methods (Least absolute shrinkage and selection operator (LASSO), Ridge, decision tree) are the three main categories of feature selection (McCombe [2019]). Various studies have been published using the methods described above as filtering (Tsamardinos et al. [2019]), wrapper (Kavakiotis et al. [2017]), and embedding (Li and Huang [2018]) for whole-genome SNP data. In addition, there have been several existing studies conducted using various algorithms including Recursive Feature Elimination

(Jeon and Oh [2020]), Genetic Algorithm (Mirjalili and Mirjalili [2019]), and Particle Swarm Optimization (Kennedy and Eberhart [1995]) to identify important features in high-dimensional data; but they are time-consuming, require high computational resources, and have an exhaustive iterative process.

To tackle the high-dimensionality challenge using plant whole-genome data, this study is narrowed down to utilize five advanced penalized methodologies (such as Ridge, LASSO, Elastic Net, Group LASSO, and SGL) and a deep learning method (BNN). The penalized models used in this study can effectively handle a large number of predictors, resulting in more accurate and efficient results. Furthermore, BNN can capture complex non-linear relationships between genetic markers and phenotypes and reduce over-fitting. The following section provides a brief introduction to the machine and deep learning models used to develop two classifiers or regressors.

1.3 Research Models: An Overview

1.3.1 Penalized Models

The penalized models (Ridge, LASSO, and their variants) generate sparse solutions which aid in feature selection. These are predictive models based on the expression of a small number of SNPs, enabling feature selection for high dimensional data (Ma and Huang [2008]). The magnitude of the penalty in penalized models can be tuned to choose the most important SNPs and other predictors. Hard thresholding reduces the dimensionality of a model by excluding variables whose coefficients are close to zero (Liu and Foygel Barber

[2020]). Ridge (Hoerl and Kennard [1970]), one of the penalized machine learning techniques, can force the least important coefficient weights close to 0 in the regularization paths. It is possible to choose a small subset of predictors from a large set of predictors, that is, reduce coefficient values exactly to 0 known as soft thresholding (Liu and Foygel Barber [2020]). LASSO (Tibshirani [1996]), Group LASSO (Yuan and Lin [2006]), Sparse group lasso (SGL) (Simon et al. [2013]), and Elastic net (Zou and Hastie [2005]) are some of those soft thresholding regularization algorithms with a varied penalty term. For instance, LASSO has l_1 penalty which is the absolute value of the magnitude of coefficients.

This study aims to combine five penalized models in two phases to develop a penalized-based workflow- PentaPen. PentaPen is a classifier and regressor developed for SNP identification to minimize the variance in the data. Additionally, there are some studies on penalized techniques for high-dimensional data been conducted recently (Gao et al. [2014], Liu et al. [2013]). For instance, various studies (Van Wieringen et al. [2009], Bøvelstad et al. [2007]) find that Ridge outperforms LASSO for prediction performance. Whereas using a simulation study (Ogutu et al. [2012]), it was found that LASSO outperforms Ridge. However, to our knowledge, no detailed analysis has been done of how the penalized models compare favorably against one another for whole-genome SNP data. Our research intends to bridge this knowledge gap, offer a thorough pathway for comparing these models' performance, report the computational time, and report the model complexity by the penalized models and PentaPen.

1.3.2 Deep Learning

1.3.2.1 Hypothesis test for preliminary screening

Hypothesis testing is a widely used statistical method in genetics research for feature selection (Huynh-Thu et al. [2012]). Researchers often use chi-square hypothesis testing to select categorical/binary variables and ANOVA (Fisher [1954]) to select continuous variables to identify genetic variants that are associated with specific diseases or traits (Galesloot et al. [2014], Zhuang et al. [2012]). In this procedure, a subset of genetic markers, such as SNPs, that are most strongly associated with the phenotype is chosen. Feature selection is important because it can help reduce the dimensionality of the data and make it easier to find the most useful genetic markers (Wu et al. [2019]). In this study, we use chi-square and ANOVA tests to select preliminary features of the binary/categorical and continuous phenotype respectively from the original data to train deep learning models. Performing preliminary feature selection is essential to reduce the number of features as this allows neural networks to provide more accurate predictions.

1.3.2.2 Models

Researchers used different deep-learning techniques for prediction and feature selection. For instance, various studies on high-dimensional data were conducted using Bayesian Neural Networks (BNNs). Neal and Zhang [2006] proposed a new method for high-dimensional classification that combines BNNs and Dirichlet diffusion trees. The authors aimed to improve classification accuracy in high-dimensional data sets by using these two techniques

together. The BNN provided a flexible and powerful model for classification, while the Dirichlet diffusion tree provided a way to handle the curse of dimensionality in high-dimensional data sets. Results showed that BNN applied with Dirichlet diffusion trees outperformed other deep learning models. The purpose of Gianola et al. [2011]'s study was to investigate the potential of some Bayesian Artificial Neural Network (ANN) architecture in predicting complex quantitative traits in two different data sets, Jersey cows and wheat. The results indicated that adding non-linearity from ANN along with Bayesian uncertainties improved the predictive ability in both data sets.

Due to the strengths of Bayesian combined with deep learning models, the objective of this research is to develop a BNN-based workflow- BayesDL - which makes predictions on selected SNPs using Chi-square/ ANOVA. BayesDL is a cascaded classifier and regressor of ANNs followed by Bayesian inference. BayesDL is developed using Stan (Carpenter et al. [2017]), a probabilistic programming language. Stan provides full Bayesian inference for various models, including linear and non-linear regression, time-series analysis, general linear models, generalized mixed models, hierarchical models, and more. The goal of Stan is to provide a simple platform for specifying complex models and doing inference using methods like variational inference (VI) and Markov Chain Monte Carlo (MCMC) algorithms. Additionally, we aim to use Stan to perform MCMC diagnostic and compare BayesDL's prediction performance with the Convolutional Neural Network (CNN). Further, BayesDL aims to output important SNPs used for follow-up biological analysis.

1.4 GWAS Software: A brief overview

GWAS is a study design used in genetics and genomics research to identify genetic variants (here SNPs) associated with physical traits. GWAS involves analyzing the entire genome of individuals to identify SNPs that are statistically associated with a particular phenotype. GWAS software refers to the computational tools and algorithms used to conduct association studies. The most popular GWAS software are PLINK (Purcell et al. [2007]), BOLT-LMM (Loh et al. [2015]), FaST-LMM (Lippert et al. [2011]), GCTA (Yang et al. [2011]), TASSEL (Bradbury et al. [2007]), and GAPIT (Lipka et al. [2012], Tang et al. [2016]). Some researchers (Fu et al. [2021], Sardos et al. [2016]) worked on comparing important SNPs output from various GWAS software. One such study (Lipka et al. [2012]) evaluated the performance of several GWAS software tools, including GAPIT, on a maize dataset. The study found that GAPIT produced highly precise and computationally efficient results compared to other tools, such as TASSEL and PLINK. But there has not been enough work to find the shared SNPs from penalized or deep learning methods compared with GWAS software. The GWAS software is designed to handle large-scale genetic data, provides a comprehensive set of algorithms, incorporates statistical models, and includes functionalities for data preprocessing. The limitations of GWAS may lead to false-positive or false-negative results.

1.5 Research Objectives, Questions, and Contributions

Given the limitations of previous research using penalized and deep learning methodologies, I aim to address the challenges and achieve the following goals displayed in Figure 1.1. The figure represents a block diagram for mapping the research objectives, questions, and expected thesis contributions.

1.6 Hypotheses

There are three main hypotheses for this study, which are mentioned below:

1. PentaPen will outperform individual penalized models in terms of evaluation metrics.
2. BayesDL will outperform the deep learning model, CNN in terms of performance metrics for prediction.
3. The identified SNPs from PentaPen and BayesDL will demonstrate limited overlap with each other and with those obtained from existing GWAS software, due to the different models used.

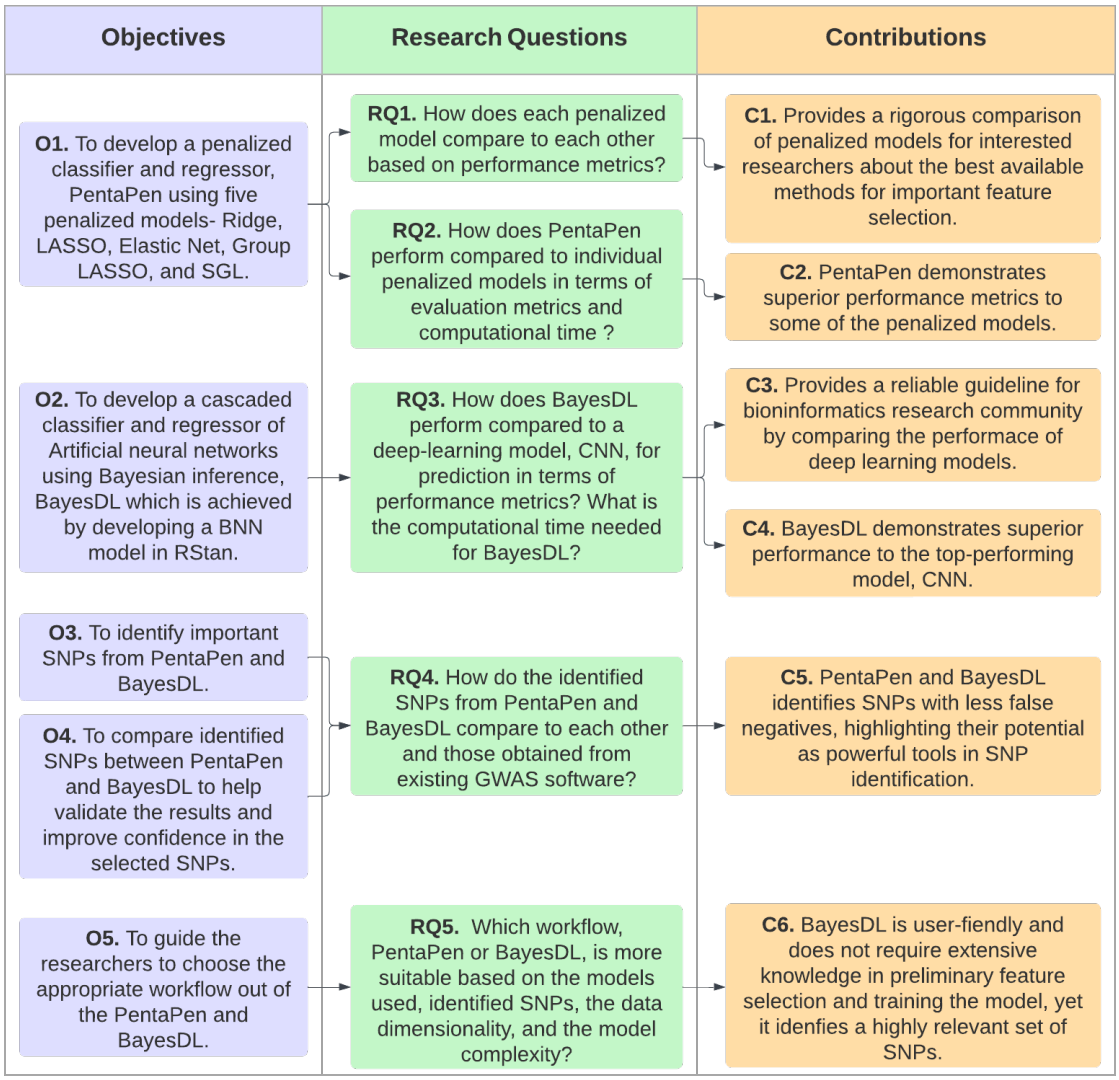


Figure 1.1: Mapping research objectives, questions, and contributions. This diagram illustrates the formulation of research objectives, and research questions, and identifying contributions to a study.

1.7 Experimental Data

To evaluate the performance of the developed workflows, two data from the model plant *Arabidopsis thaliana*, AtPolyDB and F1, are used for this study. Using the *Arabidopsis* data set for performance evaluation of the SNP analysis algorithm is justifiable as it is a widely studied model organism with a fully sequenced genome, high genetic diversity, and well-characterized SNP data sets. Developing an algorithm specific to the *Arabidopsis* genome can lead to more accurate and efficient SNP analysis. They are obtained from easyGwas website [1]. The easyGwas website is a repository of whole-genome SNP data that has been systematically processed is ready to be analyzed and is thoroughly documented. On the easyGwas website, there are presently 12 data sets including estimates for SNP counts as well as related publications for their exploratory analysis.

The AtPolyDB dataset has 1307 samples with 214051 SNPs (or features) and the F1 data set has 372 samples with 204753 SNPs. Atwell et al. [2010] used the AtPolyDB dataset for the application of GWAS to study 107 phenotypes. Seymour et al. [2016] utilized the F1 data set to study the inheritance in *Arabidopsis thaliana* hybrids. Both data sets contain three files: (a) the PED file stores genotypic data with 1307 and 372 samples followed by total fields as $6 + 2 \times p$ (p is the number of SNPs), (b) the PHENO file stores the phenotypic data having 1307 and 372 instances with 109 and 3 columns, and (c) MAP file has 214051 and 204753 rows, which contains the information about every single SNP, with four fields each row. Each row corresponds to

[1] https://easygwas.ethz.ch/data/public/dataset/view/1/ and https://easygwas.ethz.ch/data/public/dataset/view/42/

one SNP in the PED file. These are the Chromosome code, SNP(Variant) identifier, position in morgans or centimorgans, and base-pair coordinate.

Table 1.1 represents the phenotypes selected for this study. The phenotypes from the AtPolyDB data set were chosen at 22^0 Celsius to consider all the phenotypic properties of the species during the summer season. These phenotypes are commonly used in genetic studies of *A. thaliana* because of their easily observable and measurable characteristics of seed development and flowering time. These properties can affect the yield and quality of crops. Three different variable types of phenotypes are selected for this study to increase the reliability of the developed workflows.

Table 1.1: Phenotype data used in this study. Overview of Phenotype Data from AtPolyDB and F1-hybrids data sets, categorized by Variable Type.

Variable Type	**Dataset**	**Phenotype**	**Number of samples**	**Phenotype Description**
Binary	AtPolyDB	Anthocyanin_22 (AT_P_172)	177	Visual anthocyanin presence
Continuous	AtPolyDB	Width_22 (AT_P_166)	175	Plant diameter
	F1-hybrids	DTF (AT1P6701)	372	Plants were subsequently phenotyped for days to first open flower
Categorical	AtPolyDB	Germ_22 (AT_P_163)	177	Days to germination

1.8 Performance Metrics

Various quality metrics are used as there are three types of phenotypes used in this study as described in Section 3.1.2.7. Classification metrics such as Precision, Recall, F1 score, and AUC are used to evaluate the performance of classification models for the binary phenotype. Regression metrics like R-squared and RMSE are utilized to assess the goodness-of-fit of regression models for continuous phenotypes. For categorical phenotype, Accuracy is the multi-class classification metric used in this study.

1.9 Thesis Outline

The remainder of this thesis is structured as follows. The literature review is presented in Chapter 2. Chapter 3 outlines the research based on penalized and deep learning research methodologies. The findings are presented in Chapter 4. The biological interpretations, conclusion, and potential future work for this research are presented in Chapter 5. The Appendix contains several Tables of results used for the comparison and interpretation of two methodologies.

Chapter 2

Literature Review

The methodologies for GWAS, some common machine learning, deep learning, statistical approaches for SNP data, and related studies are briefly explained in this chapter. The methodology and data sets used in the published studies vary, as do the issues with whole-genome SNP data that the methods attempt to solve.

2.1 From DNA to SNP

Deoxyribonucleic acid (DNA) comprises an extremely long chain of connected single units called nucleotides: Adenine, Thymine, Guanine, and Cytosine. Species share 99% of their make-up and only 1% is responsible for the diversity observed in the individuals (Lewontin [2003]). For example, it is evident from Figure 2.1 that in a particular location says one of the 95% of individuals has A nucleotide while one of the 5% species' individuals has a T

nucleotide, each of these is called a variant. Given this location has multiple forms it is called the SNP. These are the key to understanding the genetic causes of an organism's traits (Bush and Moore [2012]). SNPs can have functional changes such as amino acid changes that cause the alteration to the mRNA transcript stability and changes to transcription factor binding affinity (Robert and Pelletier [2018]).

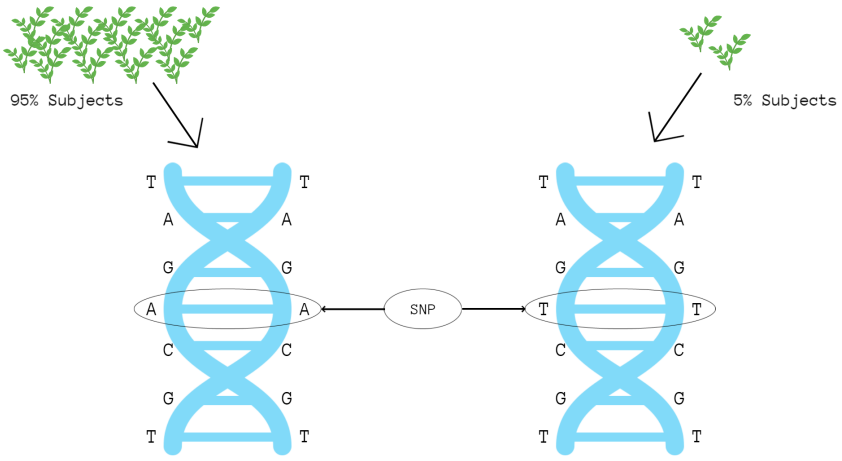


Figure 2.1: A graphical explanation of Single Nucleotide Polymorphism (SNP). The diagram illustrates the concept of SNP, a common genetic variation characterized by a single base pair change in the DNA sequence.

2.1.1 GWAS Analysis

Figure 2.2 displays the steps to be followed for GWAS analysis (Uffelmann et al. [2021]) which are described below:

(a) Data is collected to find an appropriate number of samples of individuals who differ in the trait of interest and the genotypic information is

obtained,

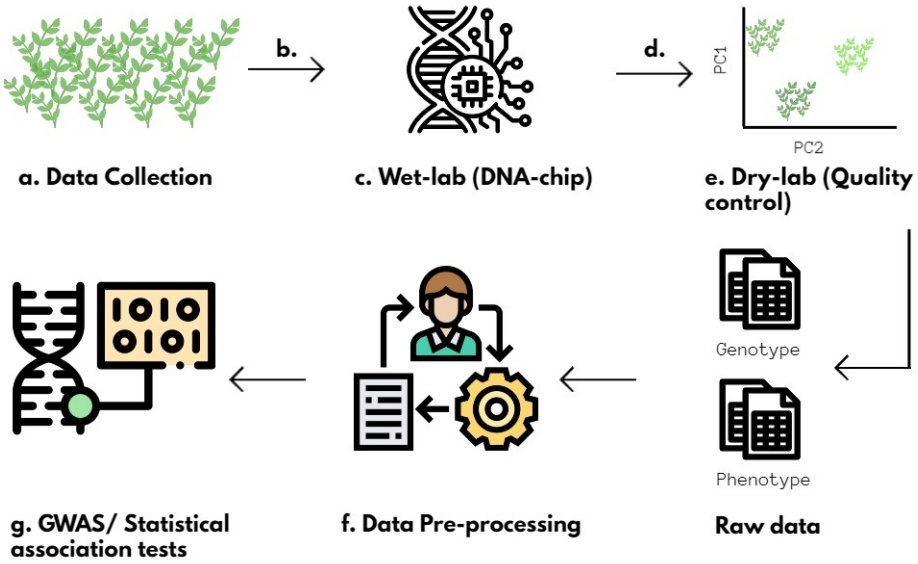


Figure 2.2: Steps for GWAS Analysis. This diagram outlines the sequential steps involved in conducting a GWAS analysis, starting from data collection (a) to obtaining final SNP results (g) for further statistical analysis.

(b) for genotyping, microarrays can be used to detect common variations or next-generation sequencing techniques can be used for whole-genome sequencing (WGS) or whole-exome sequencing (WES),

(c) the wet-laboratory procedure, such as genotype calling and DNA switches include the isolation of DNA from all the subjects using a DNA chip (a molecular kit) to identify the SNP alleles (a variant form of a gene) they have at each SNP genome position,

(d) to maximize detection sensitivity for variant calling, pipelines should include various tools (FreeBayes, GATK, Platypus, Samtools/mpileup) for each kind of variation (Koboldt [2020]); pipelines for SNP calling

rely on a reference genome. The most popular approach for this is to select one genome assembly as the standard against which all others will be measured (Leggett and MacLean [2014]),

(e) the dry-laboratory procedures on called genotypes, such as eliminating undesirable SNPs and individuals, identifying population strata in the sample, and calculating principal components involves quality control,

(f) look through all the SNPs that are examined to find SNPs where the population has different allele frequencies and SNPs with genetic differences will be marked, and

(g) the final SNP results will then be used as input data to carry out GWAS with traditional statistical methods.

Anonymized individual ID numbers, kinship relationships between individuals, sex information, phenotype data, covariates, genotype data for variants, and batch genotyping information are all included in the input files for a GWAS.

2.2 GWAS Program

Association tests can be carried out in multiple ways using statistical models. The simplest is the generalized linear model (GLM) usually used for quantitative traits and genotype classes. For control traits, logistic regression is generally preferred to predict the probability of having case status given a genotype class. This method is preferred as it allows adjustment of covariates and odds ratio as a measure of effect size (Bush and Moore [2012]).

Regression analysis is performed for every SNP in the dataset, taking each subject as a data point. The regression model of the GWAS (Cantor et al. [2010]) can be represented as:

$$\mathbf{Z} = \mathbf{X}\theta + \epsilon \tag{2.1}$$

where $\mathbf{X}$ is a $n \times (p+1)$ matrix of predictors, a $(p+1) \times 1$ vector θ consists of their coefficients, and matrix $\mathbf{Z}$ is the $n \times 1$ matrix of a phenotype. Here, n denotes an individual and p denotes a predictor (SNP). In ordinary linear regression, the variation in a trait can be fully described by two parameters - a common variance σ^2 and a mean vector in the design matrix $\mathbf{Z}$ that is unique to each individual in the sample (n). The residuals, or deviations (ϵ) from predicted trait values, are assumed to be normally distributed.

A regression line is fitted to the data that best predicts the relationship between several alleles or SNPs and a phenotype. The p-value is a statistical measure used to assess the significance of the association between the SNPs and phenotype. It measures the likelihood that the association found in the distribution of data points was due to random chance, given that there is no true association. The stronger the data points cluster around the regression line, the less likely the association is due to random chance (Uffelmann et al. [2021]). A smaller p-value indicates that the observed association is less likely to be due to random chance. For each SNP, the p-value and the slope (effect) of the regression line are recorded. However, working with millions of SNPs can be computationally expensive. After recording the p-values, a significance test is performed to determine whether the association is statistically significant. The null hypothesis is that there is no significant association between the SNPs and phenotype. A p-value threshold of 0.05 is commonly used to determine whether the association is significant or not.

If the p-value is less than 0.05, it is concluded that the association is not due to random chance and is considered to be present. Visualization is done using the Manhattan Plot (which resembles the Manhattan skyline) where each SNP position on a chromosome is represented on the x-axis and its associated p-value on the y-axis is precisely negative of the log of a p-value. Each dot represents each SNP position in each chromosome and the dot's height indicates a significant difference. Hence, dots above a certain horizontal line (threshold) represents SNPs that are significantly associated with the chosen physical trait (Ikram et al. [2010]).

2.2.1 Advantages and Limitations of GWAS

The GWAS program is advantageous as it allows for adding covariates (other effects that may affect the parameter of interest) along with genotypes. The GWAS has limitations due to genetic confounding and complex genetic architectures (Korte and Farlow [2013]). The heterogeneity of genetic origins among the individuals in association studies may lead to false-positive or false-negative results (Hu and Ziv [2008]). This can be due to the lack of replication of the association study. Since many of the phenotypes of interest in living beings may be complex (combination of genetic and environmental factors), GWAS may be unable to find the causative loci we seek. One possible solution is to reduce the number of loci associated with the trait improving the detection power. This can be done using the LASSO (Korte and Farlow [2013]).

However, with p = 0.05, there is still a chance of producing a false positive result, that 5% of significant results are still due to random chance.

Working with millions of SNPs leads us to produce thousands of false positives, muddling the results and diminishing the study's statistical power.

Furthermore, the high dimensionality of current whole-genome SNP data presents significant hurdles for computationally intensive GWAS techniques like permutation testing (Li et al. [2018a]). Some of the approaches that can be used to correct the multiple testing (more false positives over the entire GWAS analysis) are the Bonferroni correction and Permutation testing (Bush and Moore [2012]).

2.2.2 GWAS Software

PLINK (Purcell et al. [2007]), BOLT-LMM (Loh et al. [2015]), FaST-LMM (Lippert et al. [2011]), GCTA (Yang et al. [2011]), TASSEL (Bradbury et al. [2007]), GAPIT (Lipka et al. [2012], Tang et al. [2016]), and others are some of the most extensively utilized GWAS software. They are all based on standard statistical models, and many of them can be used to choose LMM, GLM, or other models.

PLINK is a free, open-source toolkit for population-based linkage analysis and GWAS. It provides a range of fundamental yet fast computations for huge biological data sets. It was one of the first GWAS software and was created for human genomes. PLINK is now regarded as a standard GWAS approach in several research disciplines.

BOLT-LMM uses a Bayesian mixed model that is said to improve statistical power and processing speed for larger data sets. Loh et al. [2015] applied BOLT-LMM to the Women's Genome Health Study (WGHS) and

collected data on nine phenotypes in 23,294 samples.

FaST-LMM proposes to scale linearly with data size in both runtime and memory by using a factorized log-likelihood function in the LMM. It may be a potential software for achieving satisfactory results with enormous data.

GCTA is a technique for analyzing complicated traits across the genome. It solves the "missing heritability" problem of human genomes by using LMM to fit the contribution of all SNPs as random effects.

Trait analysis by aSSociation, Evolution, and Linkage (TASSEL) and Genome association and prediction integrated tool (GAPIT) were created with GWAS of agricultural plants in consideration. TASSEL was originally designed and tested for maize to handle various insertions and deletions. Many other programs did not take these polymorphisms into account previously. GAPIT was created after TASSEL used comparable statistical methods to TASSEL. It can do both GWAS and Genomic Selection (GS) analyses (Goddard and Hayes [2007]). GS is a marker-assisted selection method in which breeding values are anticipated using genetic markers found throughout the genome (usually SNPs). GAPIT is regularly updated to provide the most recent techniques. As a result, it is said to produce the most precise and computationally efficient outputs.

Multiple loci mixed model (MLMM), fixed and random model circulating probability unification (FarmCPU), and Bayesian-information and linkage-disequilibrium iteratively nested keyway (BLINK) were the three multi-locus test techniques that were implemented by version 3 of GAPIT (Wang and Zhang [2021]). Additionally, two GS techniques built on CMLM—compressed BLUP (cBLUP) and SUPER BLUP (sBLUP) were implemented. These new

solutions increase GWAS statistical strength and GS prediction accuracy, processing speed, and the ability to evaluate large amounts of genomic data.

2.2.2.1 Bonferroni Correction

A multiple-comparison method known as the Bonferroni correction is used when several statistical tests are run together (Chen et al. [2021], Kaler and Purcell [2019]). It is a common approach to setting the local significance level as the global significance level (α) divided by the number of SNP variables (p). This was used in GWAS to reduce the chances of obtaining false-positive rates and to identify the important SNPs from the outputs. In this study, the GWAS software output was adjusted using the Bonferroni correction to identify the important SNPs from the experimental data sets. This correction was needed because the software output contained a different number of SNPs, making it an incompatible comparison with the output of the developed workflows.

2.2.3 Genotypes

There are mainly three common SNP data formats for the genotypes. They are categorized as PLINK, Hapmap, and VCF.

PLINK contains two types of files ped and map files. The samples are stored in rows while SNPs are stored in columns for the ped file. Table 2.1 represents the general structure of the ped file. It has a family ID, individual ID, paternal ID, maternal ID, and gender(1=male; 2=female; 0=unknown) as the first five columns. The sixth column in the ped file is generally re-

served for the phenotype data. The rest of the columns are SNPs for each chromosome[2].

Table 2.1: Data structure for ped file. Overview of PED File Structure: Family ID, Individual ID, Parental IDs, Gender, Phenotype, and SNP Information

famid	iid	fid	mid	sex	pheno	SNP1.1	SNP1.2	...
9381	9381	0	0	0	0	T	T	...
9380	9380	0	0	0	0	C	C	...
9378	9378	0	0	0	0	T	T	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

The map file has rows as SNPs and columns as SNPs' locations. Table 2.2 displays the widely used map file format. The first two columns are chromosomes and the SNP identifier of respective chromosomes, while the other two columns give the genetic distance (morgans) and base-pair position (bp units).

Table 2.2: Data structure for the map file. Overview of MAP File Structure: Chromosomes, SNP Identifiers, Genetic Distance, and Base-Pair Positions

Chr	SNP	GD	BP
1	rs6681049	0	6681049
1	rs4074137	0	4074137
1	rs6704013	0	6704013
⋮	⋮	⋮	⋮

The other format is Hapmap text-based file. The Hapmap file format is

[2] https://zzz.bwh.harvard.edu/plink/data.shtml

a table with 11 columns and one column for each genotyped sample. Each further row has all of the information associated with a specific SNP, with the first row containing the header labels of the samples (Chang [2020]). A chromosome-specific or generic Hapmap file or a generic file can be acquired to get the HapMap data format (Richard et al. [2003]). The attributes stored in any HapMap file are displayed in Table 2.3.

Table 2.3: Overview of the attributes of HapMap- a genotype file.

Attributes	Meaning
rs#	SNP identifier
alleles	SNP alleles
chrom	the chromosome of each SNP
pos	position of SNP on that particular chromosome
strand	SNP's orientation(forward (+) or backward (-)) within the DNA strand
assembly#	NCBI reference sequence assembly
center	genotyping facility that generated the genotypes
protLSID	HapMap protocol identifier
assayLSID	HapMap assay identifier used for genotyping
panelLSID	panel of individuals genotyped identifier
QCcode	quality control for all entries;
Parental#/ID#	subsequently, the list of sample names

Originally the 1000 Genomes Project created the VCF (Variant Call Format), a standardized format. Meta-information, header, and data lines comprise the VCF format. The meta-information is organized into rows, with the first one being required and containing the VCF format version. More, but not essential, lines indicating the file's origin can be added after the first row,

such as the file's creation date, source, reference, phasing, etc. The following entries are found in the information lines (##INFO) as ID, Number, Type, and Description. The header line comes after the meta-information lines, which contain all of the information and format data. The header line must include a minimum of eight fixed columns that are tab restricted. It also has a column named FORMAT, followed by the sample IDs if it has genotypic data. The following is the syntax of the information line:

```
##INFO = <ID=ID,Number="number",Type="type",Description="description">
```

After the information lines, a few lines indicate the format of each variable of the data. The syntax for the ##FORMAT line is as follows:

```
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
```

The header line after the meta-information lines contains the following columns and their respective data information as displayed in Table 2.4.

The data lines tab is limited after the header line, with '.' representing missing data (Lyon et al. [2021]).

2.3 Common Statistical and Machine Learning Algorithms

The popular Machine Learning (ML) and statistical algorithms (aside from penalized models used in this study) that have been discussed in the academic literature concerning whole-genome SNP data are covered in this section.

Table 2.4: Columns of VCF (Variant Call Format)- a genotype file

.

Columns	Meaning
#CHROM	an alphanumeric string of chromosome
POS	an integer displaying the position in the chromosome
ID	an alphanumeric string of the identification
REF	reference base(s)
ALT	alternate non-reference base(s)
QUAL	the quality score for the assertion made in ALT
FILTER PASS	passed in all filters otherwise should contain the reason
INFO	all about INFO described earlier separated by a semicolon
FORMAT	all the format IDs separated by a colon
IDs	a tab-separated list with sample identification

We have already seen that the high dimensionality of data sets like whole-genome SNP data presents a challenge. There are several dimensionality reduction methods available. There are various dimensionality reduction techniques such as principal component analysis (PCA), t-distributed stochastic neighbor embedding (t-SNE), linear discriminant analysis (LDA), Independent Component Analysis (ICA), and uniform manifold approximation and projection (UMAP) used by researchers. Nahlawi [2010] proposes a comparison of various dimensionality reduction algorithms for genetic feature selection, which identifies the most informative subset of genetic features from a large and complex set of genetic data. The algorithms compared in the study included popular dimensionality reduction techniques such as PCA, ICA, and Fast Orthogonal Search. Yan and Wang [2022] used popular dimension reduction methods such as PCA, LDA, t-SNE, UMAP, and ICA to reduce the dimensionality of large and complex omics data, such as genetic

and genomic data while preserving the most relevant information. However, these methods are not employed in this thesis because of their computational complexity.

Various ML models such as Random Forest (RF), Gradient Boosting Machine (GBM), Support Vector Machine (SVM), and Gradient Boosting Tree (GBT) are useful for searching SNPs in high-dimensional SNP data. Szymczak et al. [2009] applied RF and GBM to find important SNPs for GWAS. Mieth et al. [2016] proposed a two-step novel algorithm whose first step was to use SVM for training for determining a subset of SNPs. Behravan et al. [2018] utilized GBT followed by SVM to identify important SNPs.

2.4 Penalized Methodologies in the existing literature: A Brief Overview

In this section, we discuss the evidence that the penalized approaches for whole-genome SNP data have been used in the literature. It is included for discussion here if a published study analyses at least one penalized model. Penalized models are increasingly common for dealing with high-dimensional data. The basic concept of any penalized technique is to add a penalty term to the loss function of the model that shrinks the coefficients toward zero. This helps to select important variables and avoid over-fitting.

Several types of penalized models have been implemented in this study, such as LASSO, Ridge, Elastic Net, and Group LASSO, and their variations. LASSO and Ridge are two of the most popular methods that have been widely used in genomics research. LASSO imposes an l_1 penalty that sets

some coefficients to exactly zero, thereby performing the variable selection. Ridge imposes an l_2 penalty that shrinks the coefficients towards zero but does not set any of them exactly to zero. Elastic Net is a combination of both LASSO and Ridge. Group LASSO is another method that performs variable selection at the group level, meaning that it selects entire sets of variables together.

Guo et al. [2019] aims to improve the power to detect genetic variants associated with quantitative traits by combining the SGL and the linear mixed model (LMM). This is because the SGL can identify a small set of relevant variables, while the LMM can account for the effects of fixed and random factors on the outcome variable. SGL-LMM creates a fixed zero effect to learn the parameters of random effects using LMM and then uses SGL regularisation to estimate fixed effects.

Li et al. [2015] proposed a Bayesian Group LASSO approach for variable selections in non-parametric varying-coefficient models. Group LASSO results can be viewed as posterior mode estimates from a Bayesian perspective.

However, this thesis neither focuses on applying variations of Group LASSO (except SGL) nor combining any method with LMM. In the next section, there is a discussion of a comparison study of the existing penalized methodologies.

2.4.1 Comparative Studies of penalized methodologies

Waldmann et al. [2013] used the LASSO and the Elastic net to perform GWAS on two simulated data sets. They compare the performance of the

LASSO and the Elastic net using various performance metrics, such as the prediction accuracy, and the false positive rate, concluding that both methods should be used for analysis.

In their study, Okser et al. [2014] illustrated the similarity and differences in the behavior of various common regularized models such as LASSO, Elastic Net, and regularized-least squares (RLS) wrappers on two data sets. The findings of their study concluded that LASSO and Elastic Net showed similar prediction behavior but they suggest using a large number of variants for better performance when compared with the greedy RLS wrappers. They also found that these methods are well-suited for important SNP identification.

Srivastava and Chen [2010] compare Stochastic Search Variable Selection (SSVS), LASSO, and Elastic Net using genotype data of 60 unrelated individuals from the CEU population in the Hapmap project. Their study provides that all three methods handle data with more features and fewer samples. The results of their study demonstrate that SSVS outperforms LASSO and Elastic Net based on the quality metrics used.

To ascertain the efficiency in identifying actual causative SNPs, Zuber et al. [2012] undertake a thorough comparison research involving five advanced regression approaches to determine the effectiveness among them in finding causal SNPs. These approaches include boosting, LASSO regression, regression with NEG prior, regression with MCP penalty, variable shrinkage by CAR (correlation-adjusted marginal correlation) scores, and a univariate strategy (marginal correlation). They found that the CAR-based algorithm outperforms all competing algorithms in terms of correctly recovered causal SNPs and SNP ranking.

2.4.2 Algorithms developed based on penalized methodologies

Several alternative algorithms have been created by combining the various statistical and machine-learning methodologies, some of which are covered in the previous section. A few of these techniques are discussed below.

Hoffman et al. [2013] develop and evaluate a combined algorithm and heuristics framework incorporating penalized maximum likelihood penalty. PUMA (Penalized Unified Multiple Analysis) is a unified framework that aims in solving problems of proposed penalized multiple regression (PMR) algorithms. The challenges include computational speed, poor performance on genome-scale simulated data, and identification of too many associations for real data to be biologically plausible. This framework incorporates the penalized maximum likelihood penalties for GWAS analysis (i.e., LASSO, Adaptive LASSO, NEG, and MCP) and a penalty that has not been used for GWAS analysis before (i.e. LOG). They found that the developed framework has high performance than existing PMR methods. It also shows reliable results to increase the detection power of associations between SNPs and phenotypes.

Zuber et al. [2012] developed a novel multivariate algorithm for large-scale SNP selection utilizing CAR score regression, a promising new method for ranking biomarkers. The algorithm is likely based on statistical methods, such as penalized regression, or machine learning. The results depicted the CAR-based algorithm's superiority over existing models for finding casual SNPs.

Motivated by the researchers' development of algorithms like PUMA to detect association, this thesis aims to develop a well-established workflow combining penalized models to identify important SNPs.

2.5 Neural Networks in the Existing Literature: A Brief Overview

Various deep-learning methods like Artificial Neural Networks (ANNs), Deep Neural Networks (DNNs), and Convolutional Neural Networks (CNNs), and their variations have been applied to whole-genome SNP data to identify genetic variants such as SNPs, predict the phenotypic outcomes of SNP patterns, and understand the underlying biological mechanisms of diseases and traits. The evidence of the neural networks published in the previous studies using the whole-genome SNP data is discussed in this section. Though the use of deep learning in the current research is scarce, we include the possible studies involving neural networks.

Badré et al. [2021] compared various computational models such as Best Linear Unbiased Prediction (BLUP), DNNs, and other statistical models to estimate breast cancer polygenic risk scores (PRS). Their study findings concluded that DNNs outperformed various machine learning and statistical algorithms such as BLUP. Further interpretation of DNNs identifies important features for DNN predictions. These variants were also found to be associated with phenotype through non-linear relationships.

Liu et al. [2019] proposed a deep-learning framework using CNNs to predict quantitative traits from genetic variants such as SNPs. Additionally,

they investigated genetic contributions to traits using saliency maps. They also found that this deep learning model achieved more accurate results by bypassing the imputation of missing values and treating them as new genotypes.

In this thesis, the existing deep learning models like DNN and CNN, and their variations are not used for SNP identification. The next section provides a further comparison of various deep learning models with existing linear and non-linear models.

2.5.1 Comparison studies involving neural networks

Romagnoni et al. [2019] utilized various GBT models such as XGBoost, LightGBM, and CatBoost, together with ANNs using one or more hidden layers to identify and classify genetic markers such as SNPs. Their study found that the non-linear methods such as GBT or ANNs are complementary to each other for identifying genetic markers and have similar prediction performance.

To predict phenotypic information from SNP data, Liu et al. [2019] suggested an independent deep CNN model. They were the first to choose significant biomarkers (SNPs) from their training model using a saliency map deep learning visualization technique. The proposed framework was compared with various traditional statistical techniques, including Bayesian ridge regression (BRR), Bayesian Lasso, and BLUP. They found that their deep-learning framework outperformed these traditional models in detecting biomarkers and their interactions.

2.5.2 Algorithms developed based on neural network

Although research on neural networks is scarce, some studies focus on combining statistical and deep learning algorithms to form an algorithm to increase model performance or perform feature selection. This section discusses different algorithms applied together to improve the performance or perform comparative studies.

Waldmann [2018] aims in solving the high-dimensionality problem using Approximate Bayesian Neural Networks (ABNNs) to identify important SNPs. They evaluate the performance of ABNNs in the context of genomic prediction. The ABNN model was implemented in MXNET; it was demonstrated that the model produce greater prediction accuracy than Bayesian LASSO and genomic BLUP. This model could be used for providing information on important SNPs.

The performance of a non-parametric Bayesian approach in the form of a Bayesian neural network (BNN) for detecting causal SNPs in genetic association studies was evaluated by Beam et al. [2014]. BNNs are a type of machine learning model that incorporate Bayesian statistical methods to make predictions based on large and complex data sets. Using real and synthetic data, they found that BNNs outperform commonly used approaches for finding SNP interactions across a wide range of potential genetic links.

An algorithm for creating a single hidden layer feed-forward neural network was described in the study by Setiono and Hui [1995]. Using the quasi-Newton method, this algorithm stands out because it minimizes the series of error functions connected to the expanding network. The algorithm is quite

effective and reliable, according to experimental data.

Motivated by the algorithms developed by Waldmann [2018], Beam et al. [2014], Setiono and Hui [1995] a cascaded workflow of an ANN followed by Bayesian inference, is developed in this thesis using RStan for important SNP detection. The prediction performance of BNN-based workflow is compared with CNN's performance. The quasi-Newton method is used for optimization while predicting the genotypes in the developed workflow.

Chapter 3

Research Methodology

The chapter revolves around the two major workflows proposed in this research using Machine Learning and Deep Learning methodologies. This chapter is divided into two main sections: Machine Learning and Deep Learning for GWAS. The corresponding subsections describe the experimental research design, theory of the models, and proposed workflow.

3.1 Machine Learning for GWAS

In this section, the focus is on research that employs penalized methodologies. The section covers a range of topics, including the design of the research study, the underlying theory behind penalized classification and regression models, K-fold cross-validation, and a workflow based on penalized models (PentaPen).

3.1.1 Experimental Research Design

In this study, we develop an improved penalized-method-based workflow-PentaPen- to find important SNPs combined with different penalized models. "Penta" and "Pen" are abbreviated for "five" and "penalized models" respectively. PentaPen is a classifier and regressor which is developed for SNP identification. It aims to minimize the value of the loss function while simultaneously optimizing performance metrics. Firstly, all the SNPs of whole-genome SNP data are utilized for training Ridge, LASSO, and Elastic Net. The union of the output SNPs from these three models is taken as the selected SNPs known as SNP Pooling. Secondly, the selected SNPs are sent to train Group LASSO and SGL, and the union of the output SNPs from the two is the final output of the PentaPen. Finally, an aggregated model is developed by combining the predictions of all five penalized models; this model is used to calculate the performance metrics of PentaPen.

Since R is the programming language used in this study, pre-processing is needed to make the data compatible with penalized method packages in R. After loading genotype (.ped) and phenotype (.pheno) files obtained from the easyGwas website in R and the data is pre-processed by converting the chromosomal nucleotide to numeric values and creating a design matrix of dimension p $\times$ n. The matrix is transposed to form a matrix of the same dimensions as the genotype data. Null values in the phenotype data are removed; they are imputed with the mean across all columns for the SNPs in the genotype data. The process above results in the pre-processed data.

In pre-processing SNP data, we choose not to remove SNPs in Linkage disequilibrium (LD) (Korte and Farlow [2013]) and rare variants. Excluding

LD and rare variants can lead to losing important genetic information in certain populations, so keeping them helps provide a complete picture of genetic architecture. Although we do not check for the potential interaction effects (LD) between SNPs, keeping SNPs as is can improve model performance. Rare variants of SNPs were not checked for in our data set, but including them is beneficial as they can have a larger effect size than common variants and contribute to phenotypic traits. Although haplotype information could provide better results for imputing SNPs, it can be complex and requires more computational resources and time. According to Shi et al. [2018], IMPUTE2, a haplotype imputation software, was the most time-consuming. Additionally, haplotype information is not always available or accurate for plant genome data. Imputing with the mean is preferred for efficiency and good results when the missing proportion is low. Although, there were no missing values in the data used in this study, imputing by mean would be better for missing at random (MAR) data, and generally, the whole-genome SNP data is MAR data.

After pre-processing, the resulting clean data are split into training and testing folds. Further, for 5-fold cross-validation (CV), the models are fitted using R-packages- glmnet (Friedman et al. [2010]) for Ridge/ LASSO/ Elastic net, gglasso (Yang and Zou [2015]) for Group LASSO, and SGL (Simon et al. [2018]) for SGL. The data type for chosen phenotypes was binary, categorical, and continuous; hence, this study carries out both regression and classification using the stated methodologies. All the feature selection methodologies are utilized using the same R packages for regression and classification. The family argument is used as a binomial in the glmnet package for classification whereas the default Gaussian family is used for regression. The loss function for Group LASSO and SGL is considered as

logit for classification. However, the default least squares loss function is used for regression. These steps are followed to train the models in PentaPen.

PentaPen is implemented in-parallel across 5-folds using the fitted model to get output as follows from each penalized method: (a) the SNPs' coefficients higher than the mean absolute values are recorded to output the important SNPs based on their coefficients and (b) the prediction for the phenotype is carried out on both training and testing sets using the best λ reported using CV. The steps to choosing the best λ are explained in Section 3.1.2.6.1. Further, to compare the performance of different models, we use different metrics which are introduced in sections 3.1.2.7.1, 3.1.2.7.3, and 3.1.2.7.2 with the help of an optimal cutoff. The workflow allows the comparison of single penalized models among each other to gain insights into which model is most effective in reducing over-fitting and improving the model performance. The purpose of comparing the performance of each model among themselves is to give a deeper understanding of the strengths and weaknesses of each penalized model and to identify a more reasonable number of SNPs.

Additionally, to evaluate the performance of PentaPen, the following results are interpreted: the comparison is made between (a) the performance of the workflow against all the five penalized models using all SNPs, (b) the computational time with every single model (Ridge, LASSO, and Elastic net), and (c) the number of SNPs selected by each penalized model. Finally, the important SNPs from PentaPen are validated to locate genes. They are also compared with the SNPs output set of GAPIT and TASSEL.

The proposed workflow aims to enhance the confidence of the selected SNPs by leveraging the beneficial properties of five penalized methodologies. As a result, combining multiple penalized models can improve performance

by reducing over-fitting as compared to using one model. The workflow for SNP identification can also increase the confidence in choosing an SNP set as it is more probable to select informative SNPs than using a single penalized model. Hence, the union of SNPs from Group LASSO and SGL allows for further analysis and selection of a reduced number of SNPs since SGL's sparsity group-wise and within-group results in too few or no SNPs.

The schematic diagram in Figure 3.1 illustrates the steps in this study.

3.1.2 Methods

In this section, I introduce the five penalized models that are combined to develop PentaPen. The models that are used for detecting potentially important features (SNPs) are Ridge, LASSO, Elastic Net, Group LASSO, and SGL. The loss function of models (Ridge, LASSO, and Elastic Net) used before SNP Pooling was the same while their penalties changed from l_1, l_2, and both respectively. After SNP Pooling, the models (Group LASSO and SGL) chose features based on distinct groups. This study uses K-fold CV to find an optimal λ throughout.

3.1.2.1 Ridge

Whittaker et al. [2000] created the first proposal for using Ridge regression for prediction in quantitative genetics. The Ridge has an l_2 penalty ($||\theta||$) calculated by Euclidean distance metric as follows:

$$l_2\text{-norm} = ||\theta|| = \sum_{i=1}^{p} \theta^2. \tag{3.1}$$

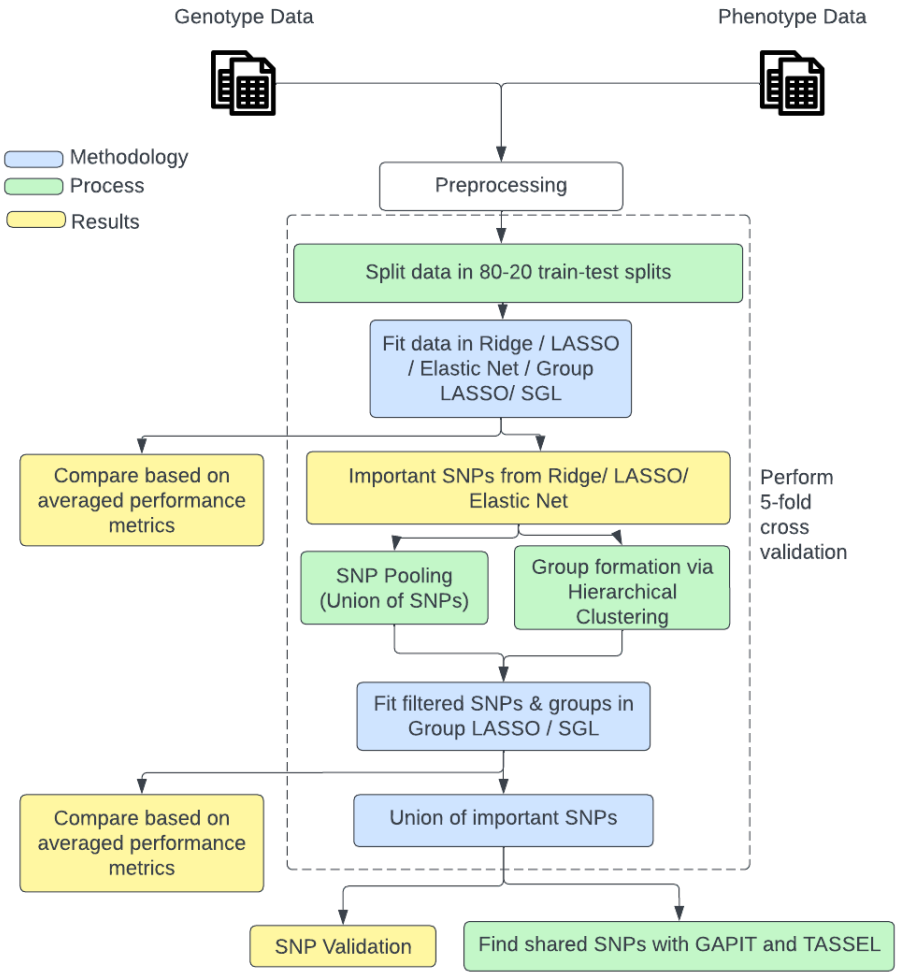


Figure 3.1: Study Design for penalized models flowchart. This flowchart illustrates the study design for penalized models starting from Pre-processing to important SNP identification and validation. The dashed box indicates the parallel computing across 5-fold.

For p predictors, matrices $\mathbf{X}_{n\times(p+1)}$ and $\mathbf{Z}_{n\times 1}$ are represented for SNPs and phenotype respectively. The objective function of Ridge (Hoerl and Kennard [1970]) with p predictors consists of the loss function $(\frac{1}{n}||Z - X\theta||^2)$ and

penalty term ($\lambda||\theta||$). Mathematically, it is represented as:

$$\underset{\theta}{\text{argmin}}(\frac{1}{n}||Z - X\theta||^2 + \lambda||\theta||) \tag{3.2}$$

where $\frac{1}{n}||Z - X\theta||^2$ is the loss function, $\lambda||\theta||$ is the penalty term, and $\lambda \geq 0$ is the tuning parameter to be estimated using cross-validation.

3.1.2.2 LASSO

The Least Absolute Selection and Shrinkage Operator (LASSO) has an l_1 penalty ($|\theta|$) calculated by the Manhattan distance metric represented as:

$$l_1\text{-norm} = |\theta| = \sum_{i=1}^{p} |\theta|. \tag{3.3}$$

Suppose the p predictors (SNPs) are arranged in the design matrix $\mathbf{X}_{n\times(p+1)}$ and phenotype is arranged as matrix $\mathbf{Z}_{n\times 1}$, then the objective function of the LASSO model (Tibshirani [1996]) for p predictors is:

$$\underset{\theta}{\text{argmin}}(\frac{1}{n}||Z - X\theta||^2 + \lambda|\theta|), \tag{3.4}$$

where $\frac{1}{n}||Z - X\theta||^2$ is the loss function, $\lambda|\theta|$ is the penalty term, and $\lambda \geq 0$ is the tuning parameter to be optimized using CV. Unlike Ridge, LASSO can produce more decipherable models by making certain coefficients exactly equal to zero.

3.1.2.3 Elastic Net

The Elastic net, proposed by Zou and Hastie [2005], combines l_1 and l_2 penalties. As a result, the Elastic net benefits from both the Ridge and

LASSO model's characteristics. The objective function of Elastic net for p predictors is:

$$\underset{\theta}{\operatorname{argmin}}(\frac{1}{n}||Z - X\theta||^2 + \lambda[\alpha|\theta| + (1-\alpha)||\theta||]), \tag{3.5}$$

where parameter α determines the mix of penalties and is chosen on qualitative grounds. A sparse model is produced by the l_1 norm portion of the penalty, while very large coefficients are shrunk by the l_2 norm portion of the penalty (Mahdi et al. [2021]).

3.1.2.4 Group LASSO

The Group LASSO model separates the predictor variables into g distinct groups. This can be useful for handling categorical data, where the groups may represent factor-level indicators. Vector θ should be approximated so that it only needs a few of those groups as opposed to just sparsity. Yuan and Lin [2006] proposed the objective function of the Group LASSO model for a symmetric and positive definite kernel matrix to estimate $\hat{\theta}$ as:

$$\underset{\theta}{\operatorname{argmin}}(||Z - \sum_{i=1}^{g}(X_i\theta_i)||^2 + \lambda \sum_{i=1}^{g}(||\theta_i^T K_i \theta_i||^{1/2})), \tag{3.6}$$

where X_i is a $n \times (p_i + 1)$ sub-matrix of X (columns corresponding to SNPs in group i) and θ_i is a sub-vector of coefficients of length $p_i + 1$; $K_i = p_i I_{pi}$ denotes the kernel matrix; the regularisation parameter is denoted as $\lambda \geq 0$.

3.1.2.5 Sparse Group Lasso (SGL)

A sparse set of groups is produced by Group LASSO where including a group in Group LASSO keeps all coefficients in the group as non-zero. For instance, when predictors are SNPs and the important SNPs have to be discovered then both sparsity of groups and within each group should be considered. Simon et al. [2013] proposed the SGL estimator model as:

$$\underset{\theta}{\operatorname{argmin}}(||Z - \sum_{i=1}^{g}(X_i\theta_i)||^2 + (1-\alpha)\lambda \sum_{i=1}^{g} ||\theta_i^T K_i \theta_i||^{1/2} + \alpha\lambda|\theta|). \tag{3.7}$$

where $\lambda \geq 0$ is the tuning parameter and $\alpha \in [0, 1]$ is a convex combination and $\alpha = 0$ gives Group LASSO model and $\alpha = 1$ gives a LASSO model. Similar to Group LASSO, SGL chooses features on a group basis, and specific features group-wise and within each selected group.

3.1.2.6 K-Fold Cross-Validation

The study employs K-fold CV to determine an optimal value of λ, which is used to evaluate the performance of the penalized models during validation and training. The purpose of it is to find a best-fit model for the database of the study and flag problems like over-fitting (Berrar [2019]). A method to train the model using a subset of the data and employing the complementary subset of the data for the model validation is known as cross-validation (CV). K-Fold CV (k-fold CV) is one of the methods of CV. In k-fold CV (Jung and Hu [2015]), all observations are randomly divided into k parts/folds of approximately equal size. At every iteration, there is a different subset reserved for testing.

3.1.2.6.1 Best λ To tune the hyper-parameter (λ) in k-fold CV, a grid of values for λ is used. The validation performance metric is calculated for each λ within each fold. Further, the overall CV performance metric is calculated for each λ. The λ with the optimized metric is located and is said to be the minimum CV λ or best λ. Model performance in this study is assessed using a 5-fold CV. The best λ considered is the λ within 1 standard deviation above the minimum λ.

3.1.2.7 Performance Metrics

This section covers the various performance metrics used for evaluating the models' performance. It can be noticed from Section 1.7 that three types of phenotypes are used in this study: binary, continuous, and categorical. Hence, there will be three types of evaluation metrics as described below to perform classification, regression, and multi-class classification.

3.1.2.7.1 Classification The binary phenotype, Anthocyanin, in our study, is imbalanced where the probability of class 1 is 36% and the probability of class 0 is 64%. To predict the presence of anthocyanin, we need to weigh the positive class or class 1 more than class 0. Hence, the cost-efficient evaluation metrics, F1 score, Precision, and Recall are used (Brownlee [2020]). Further, AUC is used to measure the classifier's ability to distinguish between both classes.

Precision measures the accuracy of positive predictions. It is the proportion of positive predicted cases that are truly positive. Mathematically,

the precision can be expressed as follows:

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{3.8}$$

where TP and FP are True Positive and False Positive respectively.

Recall measures the completeness of positive predictions. It is the proportion of all positive cases that are correctly predicted positive. It calculates the number of Actual Positives that the model detects by labeling it as Positive (True Positive). When there is a high cost associated with False Negative (FN), Recall is the model metric used to select the better model compared to other models. This metric can be mathematically written as:

$$\text{Recall} = \frac{TP}{TP + FN}. \tag{3.9}$$

The **F1 score** is the weighted average of precision and recall. The F1 score have the following formula (Goutte and Gaussier [2005]):

$$\text{F1 score} \quad = \quad 2 * \frac{\text{Precision * Recall}}{\text{Precision + Recall}}. \tag{3.10}$$

Sensitivity (or true positive rate or Recall) is the probability that a test will result in a true positive outcome. The likelihood of a negative test provided itact negative is referred to as specificity (or true negative rate). The true negative rate (TNR) and the false positive rate (FPR) are calculated as follows:

$$\text{Specificity (TNR)} = \frac{TN}{TN + FP}, \text{ and} \tag{3.11}$$

$$\text{FPR} = 1 - \text{Specificity} = \frac{FP}{TN + FP}. \tag{3.12}$$

A graphical representation of the trade-off between sensitivity and specificity at each "cut-off" for any diagnostic test is provided by the receiver operating characteristic (ROC) curve (Park et al. [2004]).

The optimum "cutoff" value with the highest sensitivity and specificity is generally accepted as the one that maximizes the area under the ROC curve (**AUC**), providing the best balance between sensitivity and specificity. AUC represents the discriminatory power of a diagnostic test (Faraggi and Reiser [2002]). AUC's computing cost, however, is greater than that of Accuracy and F1 score (Hossin and Sulaiman [2015]).

3.1.2.7.2 Regression The performance metrics R-squared and Root Mean Square Error (RMSE) were averaged across 5-folds for the continuous phenotype in this study to evaluate regression model performance.

R-squared measures how much of the variation in the dependent variable the model can account for. Mathematically, R-squared is represented as:

$$R^2 = 1 - \frac{SS_{Error}}{SS_{Total}}. \tag{3.13}$$

where $\frac{SS_{Error}}{SS_{Total}}$ is the proportion of total variation that cannot be explained by the model.

Mean Squared Error (MSE) denotes an absolute measure of the fit's goodness. MSE is a measure of the average of the squares of the errors between the predicted and actual values in a regression analysis. It is determined by adding together the squares of the differences between the real output and the predicted output and dividing the result by the total number of data points. The square root of MSE is called **Root Mean Square Error** which is abbreviated as RMSE. It can be defined in terms of the sum of the square of errors as:

$$RMSE = \sqrt{\frac{SS_{Error}}{N - p - 1}} \tag{3.14}$$

where SS_{Error} is the unexplained variability of the dependent variable.

3.1.2.7.3 Multi-class classification This study uses Accuracy averaged across 5 folds to evaluate the multi-class classification model's performance for the categorical phenotype.

Accuracy measures the true positive and true negative outcomes ratio for the entire data. It estimates the number of correct predictions over the total number of predictions. The mathematical way of representing accuracy is (Goutte and Gaussier [2005]):

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}. \tag{3.15}$$

3.1.3 PentaPen: A Comprehensive Workflow

Workflow 1 summarizes the steps to develop PentaPen, using five penalized models to detect potentially important SNPs. The data are fitted to train and validate the models as explained in Section 3.1.1. Note that the groups of SNPs that are used in Group LASSO and SGL are created using hierarchical clustering (Nielsen and Nielsen [2016]). As described in Section 3.1.1, the evaluation metrics, predictions, and selected SNPs from the models are recorded across the 5-fold.

The reasons for using SNP Pooling are as follows: the results of SGL and Group LASSO exhibit potential over-fitting and poor model performance when using all SNPs. Training Group LASSO and SGL using all the SNPs require extensive computational resources (the system runs out of memory) for some phenotypes (preliminary results not achievable and not shown here).

Workflow 1 PentaPen: Proposed Workflow

1: **Input:** Genotype .ped file and Phenotype .pheno file
2: Pre-process the data
3: **for** phenotype in phenotypes **do**
4: Initialize the folds to 1
5: **while** folds <= 5 **do**
6: Split the data into training (80%) and testing (20%) folds.
7: Train and validate Ridge, LASSO, and Elastic Net models using glmnet.
8: **for** each model **do**
9: Predict the phenotype value for both training and testing folds. Find λ within 1 standard error of the minimum obtained by an inner 5-fold CV.
10: Record the appropriate performance metric using the optimal cut-off.
11: Record the potentially important SNPs from the model.
12: **end for**
13: Filter SNPs by taking the union of SNPs from the Ridge, LASSO, and Elastic net. Create groups of SNPs using Hierarchical Clustering.
14: Utilize filtered SNPs and formed groups to train and validate Group LASSO and SGL using R functions grplasso and SGL. Repeat steps 9 and 10.
15: Take the union of the potentially important SNPs from both Group LASSO and SGL. Increment the fold number.
16: **end while**
17: **end for**
18: **Output:** The top 10 important SNPs for each phenotype.

A few improvements of PentaPen over the existing literature based on the results are stated as follows: Combining multiple penalized models leads to more reliable results by mitigating the limitations and biases of individual penalized models. It also identifies a potentially reasonable number of SNPs as the same SNPs are likely to be identified by multiple methods, thereby reducing the risk of false negatives. It improves the model's performance by addressing issues like over-fitting (Hawkins [2004]). It also improves SNP identification (due to SGL's sparsity).

3.2 Deep Learning for GWAS

This section is centered around the use of deep learning models. The various topics covered in this section include the study's design, which is based on utilizing neural networks. Additionally, the theory behind hypothesis testing and neural networks is discussed. The chapter also proposes a workflow based on a Bayesian neural network (BNN).

3.2.1 Experimental Research Design

In this study, we propose BayesDL (a BNN-based workflow) to identify important SNPs. BayesDL is a cascaded classifier and regressor of an Artificial neural network using Bayesian inference. The model is fitted using R and Stan. Firstly, the whole-genome SNP data undergoes preliminary feature (SNP) selection using hypothesis testing procedures to reduce the size of the data. It is needed to make the data compatible with neural network (NN) architectures (especially BNN which also includes probabilistic distributions)

as they require extensive computational resources (more than 32GB RAM when using all SNPs as input layer). Secondly, the BNN model is defined using Stan; the defined model is further used for SNP identification and test performance evaluation. Finally, the existing CNN model is utilized for comparison based on test performance metrics which aids in determining the superiority of BayesDL. The final output (important SNPs) from BayesDL is used for further biological analysis. The following paragraphs provide steps followed to perform preliminary feature selection and define the research work required to train and validate the NN-based models.

Before conducting analysis, it is necessary to pre-process the SNP data to ensure compatibility with the NNs used in this study. To achieve this, several steps are taken which are similar to the steps mentioned in Section 3.1.1. After completing the pre-processing of the data, preliminary feature selection is carried out using hypothesis tests (see Section 3.2.2.1). Although the independent variables (SNPs) are categorical, the dependent variable (phenotype) can be either categorical/binary or continuous. Due to this, two hypothesis tests are utilized to select the SNPs that are highly associated with the phenotype. The chi-squared test (explained in Section 3.2.2.1.1) is utilized for categorical phenotypes (with two or more categories), while ANOVA (see Section 3.2.2.1.2) is used for continuous phenotypes. The subsection below describes the stages in which preliminary feature selection is performed.

3.2.1.1 Preliminary feature selection

The preliminary feature selection from the original data is done in the following steps using an R-package, *stats* (Team et al. [2020]):

1. Perform Chi-square and ANOVA tests to evaluate the association between the response variable (phenotype) and each predictor variable (SNP) under the null hypothesis.

2. Use the test statistic of Chi-square (Eq. 3.16) and ANOVA (Eq. 3.17) to find the p-value of each association. In R, the *summary()* function and *chisq.test()$p.value* are used for ANOVA and Chi-squared respectively.

3. Find the p-th quantile (using *quantile()* function in R) and use the significance threshold appearing within the range of $0.5-2.5\%$ quantile.

4. Record the number of SNPs for various thresholds of the significance level (ranging from 0.1 to $1e-17$). The number of selected features for each phenotype, based on six different thresholds, can be found in Table 3.1.

The preliminary feature screening is preferred due to the heavy computational requirements (RAM of at least 32GB) of NNs, especially BNNs. Additionally, this is preferred as NNs perform well when the number of samples (n) is larger than the number of predictors or SNPs (p).

Table 3.1: The number of SNPs selected for the corresponding significance level threshold. The chi-square test and ANOVA are used to record these thresholds. The marked number of SNPs are finally used as input of Neural networks.

Tests	**Phenotype**	**Significance level**					
		0.10	**0.05**	**0.01**	**1e−3**	**1e−4**	**1e−17**
Chi- square	Anthocyanin	26503	15590	4580	729	**118**	–
	Germ	27930	16246	5007	991	**238**	–
ANOVA	Width	26225	13857	3182	408	**96**	–
	DTF	83074	69127	47280	29423	19154	**96**

Further, the filtered data obtained from preliminary feature selection is split into 50% training and 50% testing folds which are swapped later. Since using a 5-fold CV approach is computationally expensive for NNs specially BNNs, hence, performing a 5-fold CV is out of the scope of this study. In addition, splitting the data in half avoids the model from memorizing the training data, hence, reducing the risk of over-fitting. This splitting also ensures the model's consistency. Both the NNs are trained and validated on a train-test split and vice-versa for further analysis. The proposed BNN-based model is created using the RStan library by defining various blocks like data, model, and parameter blocks in the Stan file. These blocks are described in detail in Section 3.2.3.1. This model is first trained using the training set and the appropriate performance metrics (see Section 3.1.2.7) of the model are recorded using the testing set in R. Further, the role of the training and testing set is swapped and the BNN model is trained and validated again. Finally, BayesDL output the important SNPs by following

the steps described in Section 3.2.3.3. The top 10 SNPs are validated to identify genes and compared with the output of GAPIT and TASSEL.

The superiority of BayesDL is confirmed by comparing the performance of the BNN model with an existing NN model. We choose the CNN model for comparison; it is employed using the Keras library to perform a classification or regression task. In the training process of CNN, Convolutional 1D layers are utilized and the performance metrics are generated using a testing set. Similar to the BNN model, the train and test sets are swapped, and the performance metrics based on the test set are recorded. The recorded metrics from the two test sets of the BNN model and CNN model are used for comparison.

To make comparisons compatible this study uses the same architecture and activation functions for both the NN models: (a) The architecture of the NN models contains an input layer with p predictors, 2 hidden layers with 50 neurons in each layer, and an output layer. For CNN, an additional parameter is added in the input layer to specify the shape of the input as a $p - dimensional$ vector. To choose the optimal number of hidden layers and neurons, a common approach was employed. Initially, a relatively simple architecture with a small number of hidden layers and neurons was used, and the complexity was gradually increased based on the test performance. The selected architecture provides a good balance between efficiency and complexity as the input data is not too large. (b) The hyperbolic tangent is used in hidden layers for both classification and regression. The softmax activation function is used in the output layer for classification. For regression, the identity or linear function is used as the activation function of the output layer as this function aids to output continuous (or real) values. (c) While

the model compilation of CNN, the Adam optimizer is used for all the phenotypes while binary cross-entropy, MSE, and categorical cross-entropy are the loss functions used for the binary, continuous, and categorical phenotype.

BayesDL, used for SNP identification, increases confidence in the selected SNPs by leveraging Bayesian methods and deep learning advantages. Also, by providing evidence against the hypothesis that apriori SNPs are insignificant, the proposed workflow aids in increasing confidence in the selected set of SNPs. Unlike a CNN model, which uses a single value for the weights, BayesDL accounts for the uncertainty of those weights by using probabilistic distribution, thereby improving the confidence in the chosen SNP set. Additionally, using BayesDL is beneficial when working with small sample sizes, as it helps to reduce over-fitting, a common issue evident in CNN. Using the BNN model also results in less number of False negatives as compared to that when using CNN.

The steps in this study are depicted in the Figure 3.2 schematic diagram.

3.2.2 Methods

In this section, the focus is on the theory behind the methods used for NN-based research. The discussion covers several topics, starting with the theory of hypothesis tests that are utilized for preliminary feature selection. This is followed by an exploration of the different types of neural networks, including feed-forward and Bayesian neural networks. The section also delves into the theory behind Monte Carlo Markov Chain (MCMC) and Stan, which are important tools for Bayesian inference. Finally, the use of the Coefficient of Variation (CoV) for post-feature selection using the BNN model is explained

in detail.

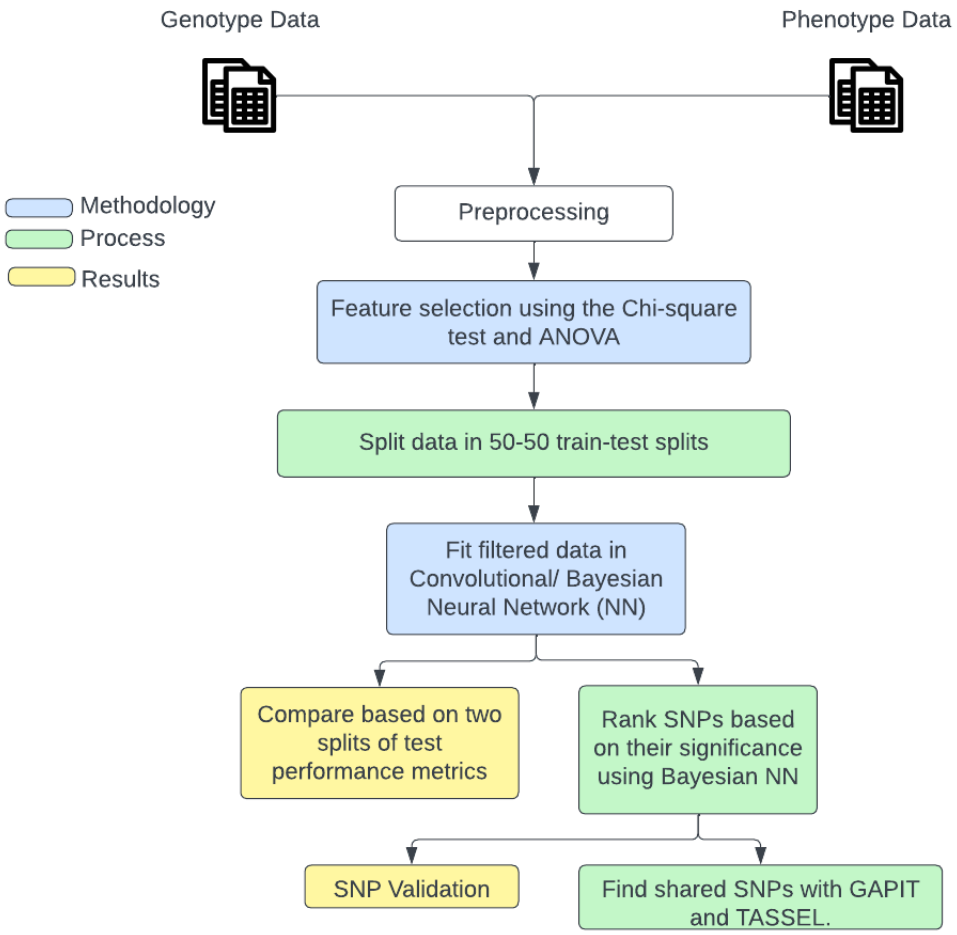


Figure 3.2: Study Design for Deep Learning flowchart. This flowchart depicts the experiment design for deep learning, outlining the key steps involved in building and training neural networks. The flowchart follows the pre-processing till important SNP identification and validation.

3.2.2.1 Leveraging Hypothesis Testing for Preliminary Screening

A statistical technique called hypothesis testing is used to assess the relationship between genetic variations (SNPs) and a certain disease or trait. It is a key technique for feature selection in genetic research because it enables researchers to identify a subset of genetic markers that are most useful in predicting the desired outcome. Several statistical methods, including machine learning algorithms, logistic regression, and linear regression, can be used to evaluate hypotheses for feature selection (Jin et al. [2018]). The goal of hypothesis testing for feature selection is to identify the most important SNPs that are associated with the phenotype of interest and can be used for further analysis or prediction. To find the genetic variants most likely to be associated with the characteristic of interest, genetic association studies sometimes employ hypothesis testing such as a t-test for feature selection (Zhou and Wang [2007]). Due to the significant computational resources (32GB RAM) required by NN models, in this research, preliminary SNPs from whole-genome SNP data are chosen using the Chi-square and ANOVA tests for binary/categorical and continuous phenotypes, respectively, to construct the input set for these models. The NN models work best with small-p-large-n data, and doing preliminary feature selection solves the problem of high-dimensional data.

3.2.2.1.1 Chi-square Test In 1900, Karl Pearson initially developed a chi-square test for testing the goodness of fit and later worked on the family of chi-square tests. The Chi-square method is a commonly used approach for selecting features when both independent and dependent variables are categorical (two or more than two). It involves evaluating SNPs on an in-

dividual basis with respect to specific classes. However, in order to apply this method to continuous-valued features, the range of values must first be divided into intervals. The Chi-square method for categorical features works by comparing the observed frequency of a class to the expected frequency of that same class, allowing for meaningful feature selection. Let O_i be the number of observed frequencies from the C_i class among the N samples. O_i should occur with expected frequency $E_i = \frac{|Ci|}{N}$. Mathematically, the Chi-squared test statistic of an SNP is represented as (Jin et al. [2006]):

$$\chi^2 = \sum_{i=1}^{C} \frac{(O_i - E_i)^2}{E_i} \tag{3.16}$$

The null hypothesis of a chi-square test is that the observed frequencies are close to the expected frequencies in a given data set. In other words, the test is used to determine if there is a significant association between two categorical variables. The larger the χ^2 value, the more informative the corresponding SNP.

3.2.2.1.2 ANOVA Fisher [1954] developed ANOVA which is a statistical technique used to determine how a quantitative dependent variable changes based on one or more independent variables that are categorical in nature. The null hypothesis of an ANOVA test is that there is no significant difference between the means of three or more groups with respect to response in a given data set. In other words, the test is used to determine if there is a significant association between the independent variable (group or category) and the dependent variable (continuous variable). The F-statistic is used to measure the significance of each predictor variable (or feature) in explaining the variation in the response variable. Mathematically, the F-

statistic is represented as (Shakeela et al. [2021]):

$$F = \frac{SSB/(p-1)}{SSE/(n-p)} \tag{3.17}$$

where SSB is the sum of squares between groups, SSE is the sum of squares within groups, p is the number of predictors (or features), and n is the sample size. A larger F-value indicates a stronger association between the predictor variables (or features) and the response variable.

3.2.2.2 Neural Networks

The neural networks (NNs) that are used for predictions are CNN and BNN. Both networks are fully connected however weights in Bayesian are assigned as probability distributions instead of a single value. To calculate the degree of uncertainty in weights and predictions, probability distributions of weights are utilized. Unlike BNN, CNN uses optimization algorithms to find the optimal set of weights and biases to optimize the loss. Instead, BNN aims to find the posterior distribution that best fits the data which is accomplished by MCMC or variational inference (VI) explained following subsections. BNN is further used for the identification of important SNPs.

3.2.2.2.1 Convolutional neural network The deep learning technique is also known as "multilayer perceptrons" (MLPs) and consists of several layers connected by neurons (Arora et al. [2015]). Figure 3.3 is a visual depiction of a fully connected feed-forward NN, CNN, consisting of several layers (an input layer, m hidden layers, and an output layer). The lines in the figure display the linked neuron which combines the weighted combinations of the inputs to generate the output parametric function. These are nonlinear

activation functions that combine linear transformations with a bias that symbolizes the activation threshold of the neuron.

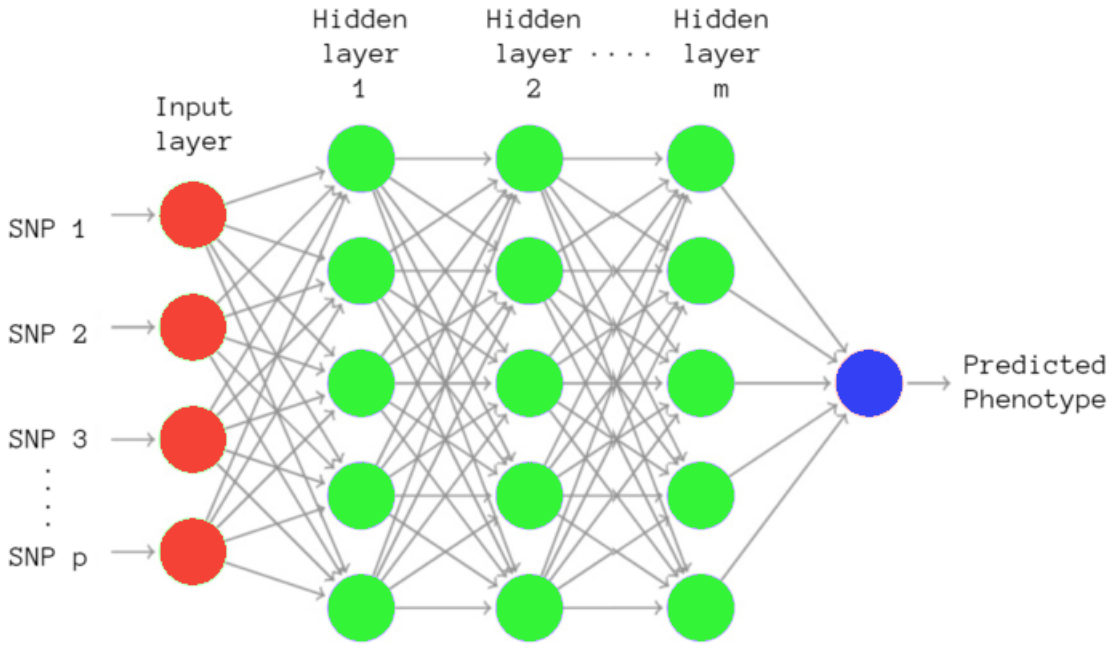


Figure 3.3: Fully Connected Feed-forward Neural Network Architecture. This diagram illustrates a neural network with an input layer, m hidden layers, and one output layer. The input, hidden layer, and output weights can be represented as θ_I, θ_{H_i}, and θ_o respectively. b_I, b_{H_i}, and b_o can represent the biases of input, hidden, and output layers respectively.

All the layers in this architecture, except the output layer (only when the response variable is standardized), contain bias. The width of the model is the number of neurons in each layer, and the depth of the model is the number of layers in the network. The output of the entire network is determined by the weights and biases connecting the neurons, but it also depends on the network's width and depth (Uppu et al. [2016]). A group of neurons,

$x_j (j = 1, \cdots, p)$, representing the input features (SNPs) make up the first layer, also referred to as the input layer. For the first layer and $i-th$ neuron, each hidden layer neuron transforms the values from the preceding layer using a weighted linear summation $c_i = \sum_{j \in input}^{p} \theta_{ji} x_j + b_0$, where θ_{ji} are corresponding weights of the particular SNP node and b_0 is the bias or the intercept of the regression equation; it is followed by the activation function $g(c_i)$ resulting in the output z_i (LeCun et al. [2015]). In matrix notation, it can be represented as:

$$\mathbf{Z}_i = g(\theta_i \mathbf{X}) \tag{3.18}$$

where $\mathbf{X}$ is a design matrix and θ_i consists of all the weights and biases. The data set $D = \{x_i, z_i\}$ is divided into training and testing sets for implementation.

The disadvantage of neural networks' extreme flexibility is that they are particularly prone to over-fitting. Over-fitting occurs when the learning algorithm performs a good enough job of optimizing the objective function to tune the model parameters for performance on the training set that the performance on new cases declines. When we select network weights that perform well on training data but badly on test data, over-fitting occurs. The issue gets worse as the network's layers are increased. Over-fitting is mitigated by Bayesian NNs as it can provide estimates of the posterior uncertainty of the model, which can be used to identify when the model is uncertain about its predictions and when it may be over-fitting to the training data (Hernández-Lobato and Adams [2015]).

3.2.2.2.2 Bayesian neural network In Bayesian inference, the objective is to determine the conditional distribution of the weights given the

training data. The conditional distribution is represented as $p(\theta_i|D_{train})$ which is also known as posterior distribution. Figure 3.4 describes the BNN consisting of an input layer with p predictors, 2 hidden layers with 50 neurons each, and an output layer that predicts a continuous, binary, or categorical phenotype. The optimal number of hidden layers and neurons is chosen by following a common strategy to start with a relatively simple architecture, with a small number of hidden layers and neurons. Gradually the number of hidden layers and neurons is increased, based on the performance of the test data until the metrics' value except RMSE starts to lie within the range of 0.80 to 1. In this study, architectures with 2 hidden layers and 50 neurons in each layer provide a good balance between efficiency and complexity as the input data is not too large or complex. The input, hidden, and output layers' weights and biases have posterior distributions that depend on the prior distributions as well.

Given the training data D_{tr}, Bayes' rule provides the posterior distribution as follows: $p(\theta_i|D_{tr}) \propto p(D_{tr}|\theta_i)p(\theta_i)$, where $p(D_{tr}|\theta_i)$ is the likelihood of training data provided by the model with parameters θ_i and $p(\theta_i)$ is the prior distribution over the parameters (weights and biases). The Bayesian model average (BMA) is then used to provide **predictions for a new test example (x_{new})** given by:

$$\mathbf{p(z_{new}|x_{new}, D_{tr}) = \int_{\theta_i} p(z_{new}|x_{new}, \theta_i)p(\theta_i|D_{tr})d\theta_i} \tag{3.19}$$

where the predictive distribution for given values of parameters θ_i is $p(z|x_{new}, \theta_i)$.

The parameters from many models are averaged together to provide the BMA estimate for thetas (or unknown parameters) (Hoff [2009]). Unfortunately, for neural networks, the BMA integral in equation 3.19 cannot be evaluated in closed form; hence, the approximate inference (for instance

MCMC; see Section 3.2.2.3) must be used. Furthermore, a complex posterior $p(\theta_i|D_{tr})$ with high dimensions makes approximation in equation 3.19 difficult. Wilson and Izmailov [2020] gives a thorough explanation of Bayesian deep learning. BNNs use BMA to average the predictions of multiple models, each with different values of the model parameters. This can help to reduce the variance of the model and improve its generalization performance, hence, reducing over-fitting (Lakshminarayanan et al. [2017], Neal et al. [2011]).

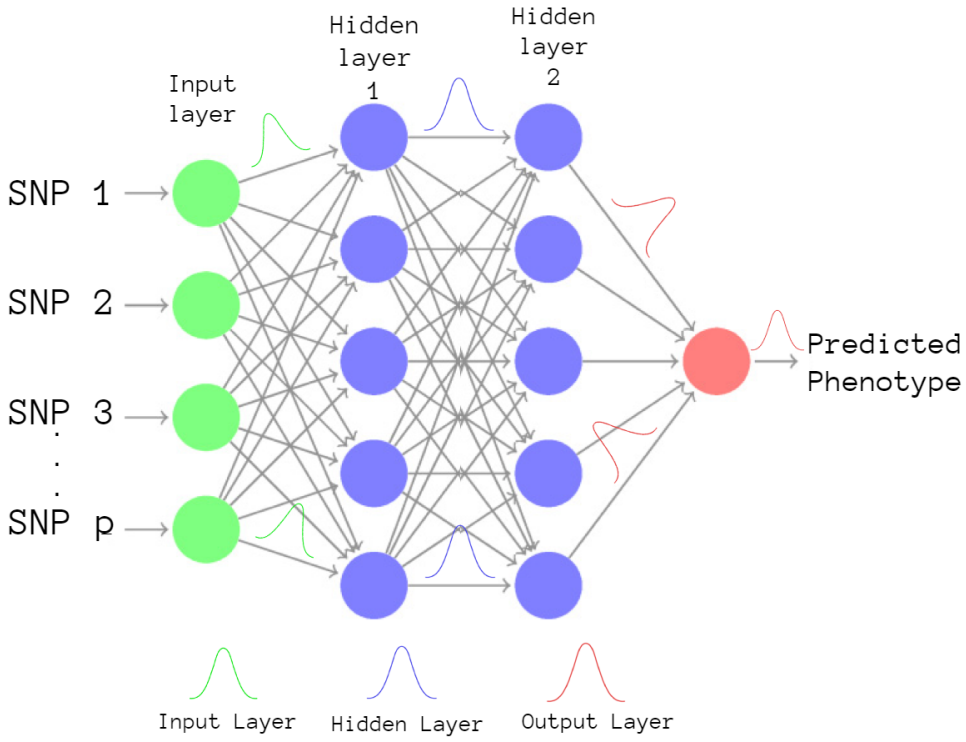


Figure 3.4: Bayesian neural network with an input layer, 2 hidden layers with 50 units each layer, and one output layer. The neural network's neurons are interconnected by lines, and each line has a weighted distribution in each layer. Additionally, the biases associated with each layer also have distributions.

3.2.2.2.3 Activation functions Hyperbolic tan (tanh) is used as the activation function for all the hidden layers. Tanh is symmetric around 0, hence scales the output values in the range $[-1, 1]$. The non-linear property of tanh captures the non-linear patterns in the data. For classification, the softmax function is used as the activation function for the output layer as it normalizes the output, ensuring that the sum of the probabilities for all classes is equal to 1. This can prevent the NN from biasing towards certain classes and help to improve its generalization performance. For regression, the linear or identity function is used as the activation function. The prior distribution for output weights is a normal distribution that is used to sample from the posterior distribution of the weights during the sampling procedure. Therefore, the output of the BNN for regression would then be computed as a linear combination of the input features and the sampled weights.

3.2.2.2.4 Priors The architecture of BNN used in this study is described in Figure 3.4. Since the weights and biases have distributions in the BNN model, hence the prior distribution for these parameters is the standard normal distribution (mean = 0, SD = 1). The standard normal distribution is one of the most widely used uninformative and conjugate prior. Moreover, the use of a standard normal prior can also help to regularize the model, since the prior effectively adds a penalty term to the likelihood function that encourages smaller weights and biases (also known as weight decay), thereby reducing the risk of over-fitting (Bishop and Nasrabadi [2006]).

The BNN model is fitted using the programming language, Stan as explained in Section 3.2.2.3.1. A similar architecture as described above is used for CNN to fit the data with weights and biases being specific values instead

of having prior distributions.

3.2.2.3 Posterior Sampling

Sampling can be used to approximate the integral in equation 3.19 by taking an average over N samples as: $p(z_{new}|x_{new}, D_{tr}) \approx \frac{1}{N}\sum_{i=1}^{N} p(y|x_{new}, \theta_i)$, where $\theta_i \sim p(\theta_i|D_{tr})$ are samples taken from the posterior. If simulated over a large number of times, a Markov chain created by Markov Chain Monte Carlo (MCMC) algorithms will produce approximations of samples from the posterior. However, in this study, we focus on the no-U-turn sampler (NUTS) (Hoffman et al. [2014]), an adaptive variant of Hamiltonian Monte Carlo (HMC) (Neal et al. [2011]), which is a generalization of the Metropolis algorithm(MA). HMC is a technique that generates samples that are asymptotically exact provided that the unnormalized posterior density $p(D_{tr}|\theta_i)p(\theta_i)$ and its gradient is available (Betancourt [2017]). The MA basic steps are to start somewhere, then propose a value nearby the current value of the iteration from the (symmetric) distribution called the proposal distribution, and then compute the acceptance ratio (or the posterior likelihood ratio = r).

3.2.2.3.1 Software Used A probabilistic programming language called Stan provides full Bayesian inference for a variety of models (Carpenter et al. [2017]). It works by allowing users to construct a probabilistic model using a high-level syntax, and then perform Bayesian inference on that model using MCMC or Variational Inference (VI) (Blei et al. [2017]).

MCMC is a simulation-based approach for roughly approximating a model's

posterior distribution. The goal of MCMC is to produce a series of samples from the posterior distribution where each sample depends on the sample before it. The NUTS, an effective and automatically tuned variant of the HMC method, is used in Stan to perform MCMC. HMC is a gradient-based MCMC technique that effectively explores the posterior distribution without getting stuck in local modes by using gradient information. A self-tuning variation of HMC called NUTS dynamically modifies the step size and direction while sampling actually occurs (Hoffman et al. [2014]).

Conversely, VI uses a simpler distribution, like a Gaussian, to approximate the posterior distribution and then optimizes its parameters to reduce the Kullback-Leibler divergence from the true posterior (Blei et al. [2017]).

Additionally, Stan offers powerful tools for model verification, diagnostics, and visualization, enabling users to comprehend the outcomes of their investigation on a deeper level.

3.2.2.4 SNP Identification using Post-Feature Selection

In this study, the coefficient of variation (CoV) is used for SNP identification using the corresponding input weights of the BNN model. It is a useful tool for feature selection in machine learning and statistics. The CoV is a statistical measure that expresses the ratio of the standard deviation (SD) to the mean of a data set. Mathematically, it is represented as (Brown [1998]):

$$CoV = \frac{SD}{mean} \tag{3.20}$$

It is commonly used in fields such as biology, engineering, and finance to measure the variability or dispersion of a data set relative to its mean value

(Bedeian and Mossholder [2000]). A low CoV indicates that the data points are clustered tightly around the mean, while a high CoV indicates that the data points are spread out widely around the mean.

The basic idea is that it can be used to measure the variability of a feature across different samples or observations. If a feature has high variability, it may not be informative for the prediction task and can be safely removed from the feature set. Conversely, if a feature has low variability, it may be informative and should be kept in the feature set (Ertuğrul and Tağluk [2017]). Hence, the minimum CoV is used to define the best feature.

One of the advantages of using CoV is that it is a relative measure that is independent of the scale of the data set. One of the disadvantages of using CoV is that it is misleading in cases of negative values and zero. To address the limitations of using CoV, we have implemented a solution in this study. Specifically, we have excluded weights with a mean of zero and calculated the absolute mean to mitigate any misleading results that could arise from negative values or zero. So, mathematically, the CoV of input weights can be represented as:

$$CoV = \frac{SD}{|mean|} \tag{3.21}$$

where $mean \neq 0$.

3.2.2.5 Performance Metrics

As discussed in Section 3.1.2.7 various evaluation metrics are used as there are three types of phenotypes used in this study. For binary phenotypes, classification metrics such as Precision, Recall, F1 score, and AUC are commonly used to evaluate the performance of classification models. On the other hand,

for continuous phenotypes, regression metrics like R-squared and RMSE are utilized to assess the goodness-of-fit of regression models. Accuracy is the multi-class classification metric used for the categorical phenotype.

3.2.3 BayesDL: A Comprehensive Workflow

Workflow 2 summarizes the steps to develop a Bayesian neural network or the proposed workflow. Load .ped and .pheno data files in R and pre-process both loaded data following the steps from Section 3.1.1. The pre-processed data is used for preliminary feature (SNP) selection which is performed using Chi-square and ANOVA tests as explained in 3.2.1.1. The data that has been filtered is divided equally into two sets - the training set and the testing set. These sets are then interchanged so that the set that was originally used for testing is now used for training and vice versa. This data set is used to train and validate the developed model. To develop the BNN model, two separate .stan files are created for classification and regression because the study includes binary, continuous, and categorical phenotypes. This BNN model is developed in Stan while R is used for prediction/ classification, sampling, and SNP identification. The following subsections describe the tasks carried out in Stan and R to develop BayesDL, and the steps followed for SNP identification.

Workflow 2 BayesDL: Proposed Workflow

Input: Genotype .ped file and Phenotype .pheno file
Pre-process the data
if categorical **then**
    feature selection using Chi-square test
else
    feature selection using ANOVA
end if
Split the data into training (50%) and testing (50%) folds.
Devolop *Stan* model using various blocks.
if categorical **then**
    stan_class = stan_model("nn_class.stan")
    Define a function to compile stan_class.
else
    stan_reg = stan_model("nn_reg.stan")
    Define a function to compile stan_reg.
end if
Call Stan's optimizing and NUTs sampler to obtain point estimates and posterior samples from the compiled model.
for phenotype in phenotypes **do**
    Fit train and test sets in the compiled model using 2 hidden layers with 50 neurons in each layer.
    Compute and record test performance metrics.
    Repeat the above process by reversing the roles of train and test sets.
end for
Rank the input SNPs by generating the samples of weights (weights ranked in increasing value of CoV) corresponding to predictors using the sampling function.
Output: The top 10 important SNPs for each phenotype.

3.2.3.1 Stan: Model Specification

Since **Stan** consists of separate blocks for defining the model, to develop BayesDL, we use seven blocks for classification and six blocks for regression. The regression model has data, parameters, function, transformed parameters, model, and generated quantities block except for the transformed data block.

The *data block* defines the parameters as the number of training and testing samples and predictors, feature matrix, outcome integer labels for classification/ vector for regression, the number of hidden layers, and the number of nodes in each hidden layer.

The *transformed data block* for classification consists of an integer value of the number of output labels. All the weights and biases of each layer along with the row vector of the number of labels for classification and a real number σ for regression are defined in the *parameter block*.

The *function block* comprises m hidden layers, each of which has the hyperbolic tangent (tanh) activation function. The last output layer is output by this prediction function block. For the final output of the output layer's vector, use the function from the function block to fit the necessary data and parameter values in the *transformed parameter block*.

The *model block* is used to define the prior distributions and the distribution for the output phenotype. All the weights and biases have a standard normal distribution for both regression and classification. For classification, the distribution of the outcome labels is modeled using a GLM with a categorical likelihood and a logit link function. This function is a softmax function

supported by Stan. For regression, the outcome is distributed normally with the mean of predicted output values and standard deviation as σ. Hence, this function is the identity/ linear activation function supported by Stan.

Finally, the *generated quantities block* contains all the test outputs generated using the function model. The softmax and normal variate are used to generate the output for classification and regression respectively.

The two files are saved as nn_reg.stan and nn_class.stan which are further read in R using the RStan library.

3.2.3.2 R: From Sampling to Inference

Now the output files from the above section are used in **R** to specify the Stan BNN model using the stan_model object (Gelman et al. [2015]). The steps described below are followed to deploy the defined model.

Create a function that builds the appropriate model and produces the desired result. This is done by defining the stan data in the function and calling the optimization and sampling methods. The optimization helps to obtain Precision, Recall, F1-score, AUC for classification, Accuracy for multi-class classification, and R-squared, RMSE for regression. The sampling method aids in drawing posterior samples from the model and selecting important features or SNPs using the corresponding weights of each predictor.

After developing BayesDL, the model was trained on the train and test sets using two hidden layers, each consisting of 50 neurons, to make predictions. Record the two test performance metrics from the optimizing method for each phenotype. The sampled weights corresponding to each node be-

tween all the layers are recorded from the sampling method and are further utilized for important SNP selection using the steps in the following section.

3.2.3.3 SNP Identification

The following steps are followed to rank the SNPs according to their significance for both regression and classification:

1. Extract all the sampled weights from NUTs sampling corresponding to the predictor.
2. Compute the posterior SD and posterior mean of the weights corresponding to each predictor across all samples or observations.
3. Remove all the weights whose posterior mean value is equal to zero.
4. Calculate the CoV for each sampled weight using equation 3.21.
5. Rank the weights based on their CoV values. Weights with lower CoV values are ranked higher and considered more informative.
6. Extract the top 10 SNPs corresponding to the weights ranked based on the lowest CoV.

Based on this analysis, the top 10 SNPs are selected from the BNN model and are used for further analysis. The pairs plot, trace plot, and auto-correlation function plot are used for MCMC diagnosis to assess the significance of the top 2 output SNPs using the *shinystan* R package (Gabry [2020]). The selected top 10 SNPs are used to identify genes and are compared with the output of GWAS software to draw biological interpretations.

This chapter provided a detailed description of penalized-based and NN-based research, including the research design for implementing penalized and NN models, the theory of different penalized and NN models used, and a detailed description of the workflow with improvements.

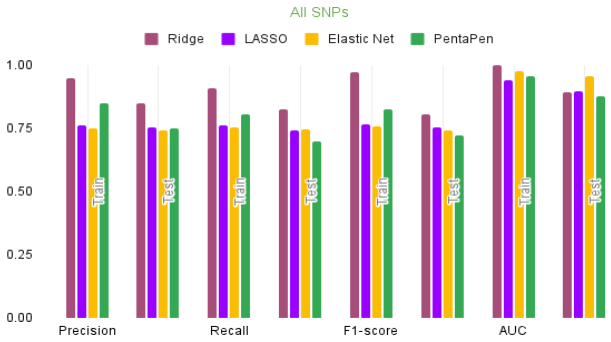
Chapter 4

Results

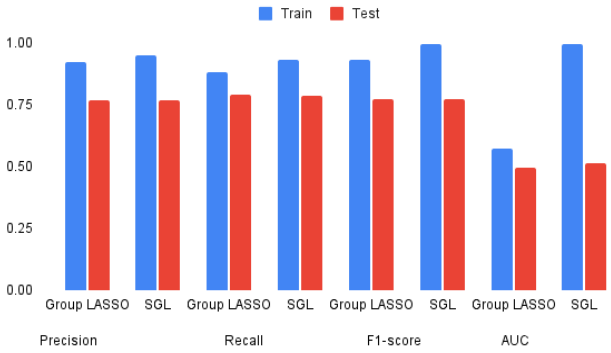
This chapter presents detailed results divided according to two major research methodologies: Machine and Deep Learning for GWAS. The following subsections within the two sections provide results based on three types of phenotypes: binary, continuous, and categorical.

4.1 Machine Learning for GWAS

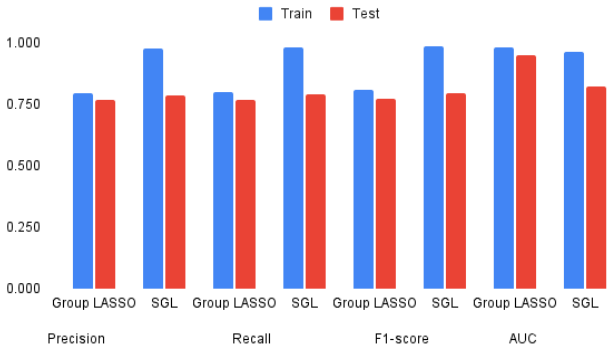
This study evaluates the performance of five penalized models and PentaPen by assessing their metrics displayed in Figures 4.1, 4.2, and 4.3. These figures are formed using Tables A.1, A.2, and A.3 which show the exact values of performance metrics in the Appendix A. Further, the findings from these figures are explained in subsections 4.1.1, 4.1.2, 4.1.3, and 4.1.4.



(a) Ridge, LASSO, Elastic Net, and PentaPen using all SNPs as input.

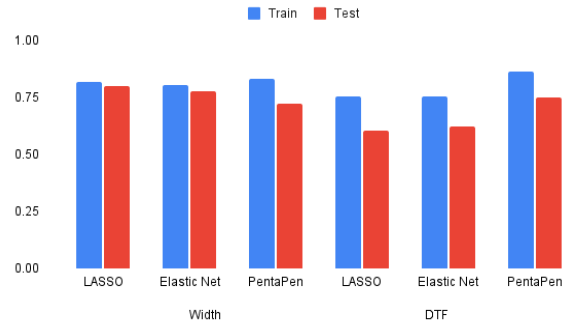


(b) Group LASSO and SGL using all SNPs as input.

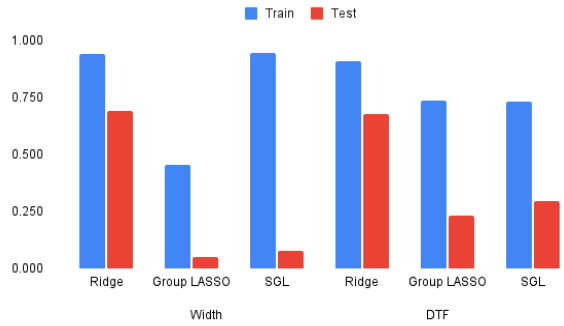


(c) Group LASSO and SGL using pooled SNPs as input.

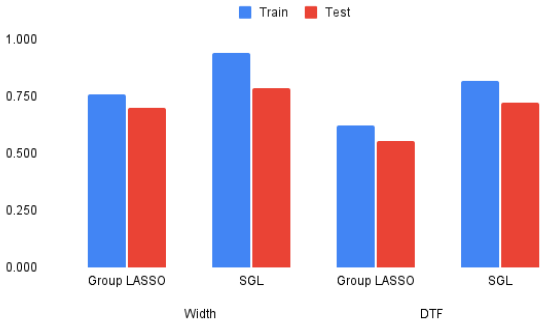
Figure 4.1: Comparison of five penalized models among themselves and with PentaPen using all SNPs as predictors for the binary phenotype, Anthocyanin. Comparison of Group LASSO and SGL among themselves using pooled SNPs. The performance metrics are recorded for both training and testing sets.



(a) LASSO, Elastic Net, and PentaPen using all SNPs as input.

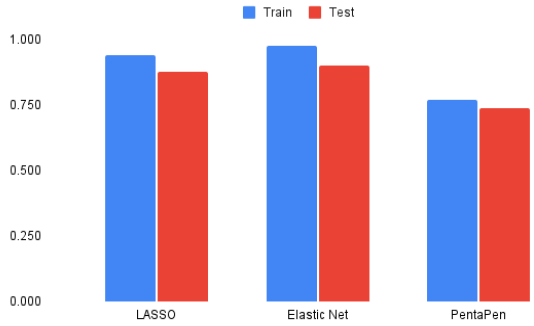


(b) Ridge, Group LASSO, and SGL using all SNPs as input.

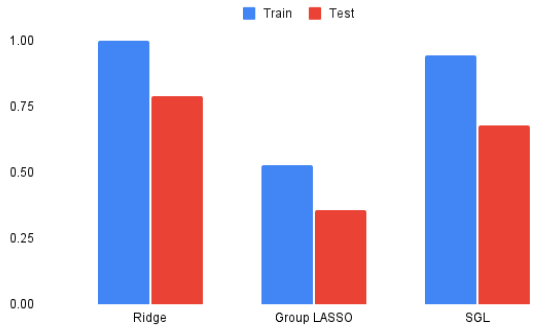


(c) Group LASSO and SGL using pooled SNPs as input.

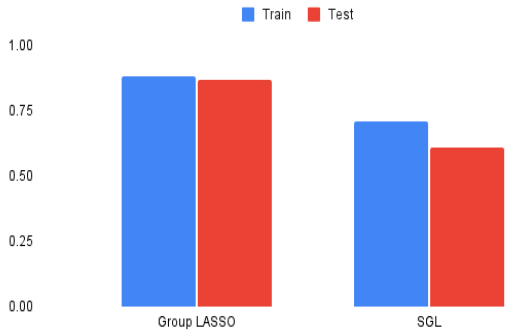
Figure 4.2: Comparison of five penalized models among themselves and with PentaPen using all SNPs as predictors for the continuous phenotypes, Width and DTF. Comparison of Group LASSO and SGL among themselves using pooled SNPs. The performance metric, R-squared, displayed in this figure is recorded for both training and testing sets. The evaluation using RMSE can be done using Tables A.1, A.2, and A.3 in Appendix.



(a) LASSO, Elastic Net, and PentaPen using all SNPs as input.



(b) Ridge, Group LASSO, and SGL using all SNPs as input.



(c) Group LASSO and SGL using pooled SNPs as input.

Figure 4.3: Comparison of five penalized models among themselves and with PentaPen using all SNPs as predictors for the categorical phenotype, Germination Days. Comparison of Group LASSO and SGL among themselves using pooled SNPs. Accuracy is recorded for both training and testing sets.

One of the objectives of this research is to systematically compare each penalized model based on input SNPs to gain insights into their strengths and limitations before including them in the proposed workflow.

Further, after forming PentaPen, the superiority of that workflow is eval-

uated using (a) the performance of the workflow against all five penalized models using all SNPs, (b) the number of SNPs selected by each penalized method, and (c) the computational time with every single model (Ridge, LASSO, and Elastic net). The important SNPs and computational time for each physical trait observed in *A. Thaliana* by the penalized models and PentaPen are displayed in Tables 4.1 and 4.2. The subsections are organized by phenotype type to explain the results in detail.

Table 4.1: Number of important SNPs selected by penalized models, SNP Pooling, and PentaPen (the proposed workflow). The SNP Pool was used as the input for Group LASSO and SGL in the proposed workflow using penalized models.

Phenotypes	**Ridge**	**LASSO**	**Elastic Net**	**SNP Pooling**	**PentaPen**
Anthocyanin	77690	77267	77387	90565	58
Width	74660	2	12	74660	4
DTF	77665	80	102	77671	77
Germ	74606	2	16	74606	5

Table 4.2: Computation time (in seconds) of penalized methods and the workflow using penalized methodologies

Phenotypes	**Ridge**	**LASSO**	**Elastic Net**	**PentaPen**
Anthocyanin	130	25	26	161
Width	109	29	28	142
DTF	104	79	51	169
Germ	113	27	24	144

When comparing the results of PentaPen, with GWAS software, GAPIT, and TASSEL were used. The Bonferroni correction was utilized in GWAS software output to find the important SNPs for AtPolyDB and F1-hybrids data. With the Bonferroni correction, SNPs were selected ranging from $13 - 97$. To make the comparison feasible and consistent among all the phenotypes, the top 10 SNPs are compared with the results of the proposed workflow. Tables 4.3 and 4.4 consist of the top 10 SNPs selected from GAPIT and TASSEL analysis for all the phenotypes for this research. The results for comparison are discussed in the subsections 4.1.1, 4.1.2, and 4.1.3.

Table 4.3: Top 10 SNPs for all the phenotypes using GAPIT. The highlighted SNPs are shared SNPs with TASSEL except DTF and Germ having no shared SNPs.

Anthocyanin	**Width**	**DTF**	**Germ**
Chr1_16930622	**Chr4_17569323**	Chr4_7749720	Chr3_11018956
Chr1_14007403	**Chr3_15742125**	Chr4_7749144	Chr4_567760
Chr4_1503923	Chr4_17569775	Chr4_7748780	Chr3_11018472
Chr1_14051264	Chr4_17570680	Chr4_5208011	Chr2_8448286
Chr1_14142053	Chr3_11052540	Chr4_5208139	Chr2_8466240
Chr1_14053491	Chr3_3504771	Chr3_7164713	Chr1_17371303
Chr1_14028301	Chr4_17572283	Chr3_7164675	Chr5_7147445
Chr1_14015352	Chr4_17571262	Chr3_17869230	Chr4_6786088
Chr2_18811115	Chr3_3687421	Chr3_5693257	Chr1_6868571
Chr3_2602503	Chr3_16092490	Chr1_1327977	Chr5_6100537

Table 4.4: Top 10 SNPs for all the phenotypes using TASSEL. The highlighted SNPs are shared SNPs with GAPIT with the exception of DTF and Germ having no shared SNPs.

Anthocyanin	**Width**	**DTF**	**Germ**
Chr1_12663142	Chr1_4636782	Chr4_10151382	Chr3_11018956
Chr1_16930622	Chr1_4637778	Chr4_12166531	Chr5_18308060
Chr1_14028301	**Chr3_15742125**	Chr4_12166556	Chr5_24425444
Chr1_14053491	**Chr4_17569323**	Chr3_2176300	Chr3_11018472
Chr1_14142053	Chr1_7431064	Chr3_2176302	Chr1_17371303
Chr1_8722354	Chr4_17570680	Chr1_22979953	Chr5_9414526
Chr1_14007403	Chr1_9804463	Chr2_526009	Chr4_8590325
Chr2_18677631	Chr4_17569775	Chr2_526254	Chr1_21874586
Chr1_13378174	Chr3_11052540	Chr1_23692187	Chr5_24431218
Chr1_19430888	Chr4_17572283	Chr1_5552218	Chr5_24432490

We further validate the results of PentaPen to locate genes from the SNPs. This was carried out by locating the corresponding chromosome and its base pair position of the particular SNP. The corresponding gene was then recorded using the Gene Model from the TAIR website. One instance of the identified SNP and its corresponding gene can be represented from the TAIR website [3]. The Tables 4.5, 4.6, 4.7, and 4.8 show the top 10 SNPs selected from the proposed workflow for each phenotype with their Chromosome base pair positions, corresponding gene, and gene function. The findings from these Tables are discussed in the following subsections organized based on phenotypes.

4.1.1 Binary Phenotype

The performance metrics of Anthocyanin to perform classification for train and test sets are Precision, Recall, F1-score, and AUC. The performance of the model enhances with the increase in the value of performance metrics. However, the wider gap between the values in the train and the test sets suggests over-fitting in the model. Although there is no generally acceptable gap that shows over-fitting in the model, in this study anything above 10% difference is considered as potential over-fitting due to the lesser samples of the data set (Park et al. [2021]). From Figure 4.1, the observations are discussed while the exact numbers are listed from the Tables A.1, A.2, and A.3 in the Appendix.

From Figures 4.1a and 4.1b (Tables A.1 and A.2 in the Appendix), it can

[3] https://jbrowse.arabidopsis.org/index.html?data=Araport11&loc=Chr1%3A1217530..1221059&tracks=TAIR10_genome%2CA11-GL-Oct22%2CA11-PC-Oct22%2CSALK_tDNAs&highlight=.

be observed that among the five penalized models, Ridge, Group LASSO, and SGL have larger Precision, Recall, and F1-score values (0.88 to 0.99 for train and 0.77 to 0.85 for the test). However, the larger gap (0.11 to 0.23) between the train and test set shows potential over-fitting in Group LASSO and SGL. For Ridge, the gap (0.07 to 0.10) does not show much evidence of over-fitting. For LASSO and Elastic net, the Precision, Recall, and F1-score values (0.74 to 0.77 for both train and test) and gap (0.006 to 0.019) between them for train and test set are almost similar. Therefore, we think that both models perform similarly based on the evaluation metrics and no evidence of over-fitting. The AUC values were larger (1 for train and 0.895 for test) for Ridge as compared with that for LASSO and Elastic net (having values as 0.942 & 0.977 for train and 0.897 & 0.959 for the test) but the difference was also more (0.10) for Ridge than LASSO and Elastic net (0.018 and 0.045). The test AUC values for Group LASSO (0.499) and SGL (0.516) show that the models are unsatisfied classifiers whereas Ridge's AUC score depicts that it is a good classifier.

Finally, PentaPen has similar Precision (0.849 for the train and 0.75 for the test), Recall (0.806 and 0.701 for the train and test), and F1-score (0.827 and 0.725 for the train and test) to LASSO and Elastic net using all SNPs. It also shows a similar gap (0.09 to 0.10) to Ridge using all SNPs between train and test sets while a smaller gap than Group LASSO and SGL using all SNPs. The proposed workflow with a high AUC score for both its training and testing phases indicates its capability to classify data accurately. Further, the PentaPen reduces potential over-fitting by decreasing the gap when compared with Group LASSO and SGL using all SNPs.

From Table 4.1, for instance, for Anthocyanin, Ridge accounts for **77690**

SNPs out of all the SNPs before SNP pooling. This might be because the Ridge evaluates each SNP individually. For binary phenotype (Anthocyanin), LASSO and Elastic Net record a similar number of SNPs as Ridge because the SNPs may be more correlated to the phenotype. PentaPen produces **58** SNPs for Anthocyanin. Ridge, LASSO, and Elastic Net identifies SNPs in the range of 77000 to 77700. However, due to the sparsity in SGL, PentaPen identifies a reduced number of important SNPs.

According to the results from Tables 4.3 and 4.5, the binary phenotype did not show any shared SNPs but had a shared gene between GAPIT and PentaPen's output. PentaPen (Table 4.5) had one shared gene, **AT3G08970**, with GAPIT. Tables 4.4 and 4.5 find that the binary phenotype had no shared SNPs or genes when TASSEL and PentaPen's outputs are compared. This could be due to PentaPen using five different penalized models to find reasonable numbers of important SNPs whereas GAPIT and TASSEL use GLM (Nelder and Wedderburn [1972]). When looking at the two GWAS programs, it showed that TASSEL had 4 shared SNPs (**Chr1_16930622, Chr1_14007403, Chr1_14142053, Chr1_14053491**) and a shared gene (**AT1G36990**) with GAPIT. PentaPen selected an SNP through the classification model with a gene, **AT1G56650**, which could be potentially related to Anthocyanin (presence of color in plants (Chen et al. [2022])).

4.1.2 Continuous Phenotype

The performance metrics used for the continuous phenotypes to perform regression are R-squared and RMSE. The model's performance is considered better when the difference between the train and test data is minimal, and

the values of R-squared and RMSE are higher and lower, respectively. From Figure 4.2, the observations are discussed while the exact numbers are listed from the Tables A.1, A.2, and A.3 in the Appendix.

It was observed from Figures 4.2a and 4.2b (Tables A.1 and A.2 in Appendix) that for Width, although Ridge and SGL have higher R-squared train values (0.944 & 0.948) and with low test values (0.694 & 0.081) the difference between their values (0.25 & 0.87) is large. This shows potential over-fitting, leaving Ridge and SGL out of comparison. Although Ridge shows over-fitting, the gap of SGL is larger than Ridge; hence, Ridge over-fits less than SGL. Comparing the other three models, Group LASSO has a low R-squared value (0.455 for train and 0.052 for testing), also showing over-fitting. Hence, comparing rest two models, both have approximately the same R-squared values of (0.82 & 0.808 for the train and 0.801 & 0.778 for the test), with a smaller difference (0.019 & 0.03). Using the RMSE values for analysis, it is evident from Figures 4.2a and 4.2b (Tables A.1 and A.2 in Appendix) that although Ridge has smaller values of RMSE but shows potential over-fitting due to a higher difference in train and test values with an exception of DTF from the F1-hybrids data set. Both LASSO and Elastic Net have smaller values for train and testing with a smaller gap between them. Using these results it can be concluded that LASSO and Elastic Net perform similarly.

Similarly, for DTF, the superiority of PentaPen is evident from Figures 4.2a and 4.2b (Tables A.1 and A.2 in Appendix) that the R-squared values (0.866 and 0.753 for train and test sets) are higher than that of LASSO and Elastic net using all SNPs with the smaller gap (0.113) as compared with the models (Ridge (0.23), Group LASSO (0.503), and SGL (0.436) using all

SNPs) that posses potential over-fitting. Similarly, when compared with each penalized model using all SNPs, RMSE values for PentaPen were lower and the workflow showed reduced signs of over-fitting. Hence, PentaPen tends to perform better for both phenotypes than models possessing over-fitting and inhibits the beneficial properties of all the five penalized models.

It is evident from Table 4.1 that for Width and DTF, Ridge accounts **74660** and **77665** SNPs out of all the SNPs before SNP pooling. For Width, LASSO records too few SNPs (**2**), whereas for DTF, **80** SNPs were recorded. Elastic net records **12 & 102** unique SNPs. Since different phenotypes may be influenced by different sets of key genetic variants leading to a smaller subset of SNPs for Width when compared with DTF. Unlike binary phenotype, the SNPs associated with continuous phenotypes may exhibit weaker correlations, allowing LASSO and Elastic Net to identify a smaller subset of relevant SNPs. Finally, PentaPen produces **4** SNPs, the union of SNPs selected by Group LASSO (3) and SGL (1).

For the continuous phenotype (Width), PentaPen showed neither shared SNPs nor shared genes with GAPIT from Tables 4.3 and 4.6. A similar result was noted between TASSEL and PentaPen from Tables 4.4 and 4.6. From Tables 4.3 and 4.4, there were 2 shared SNPs (**Chr4_17569323, Chr3_15742125**) and 4 shared genes (**AT3G43890, AT4G37370, AT3G29075, AT4G37380**) when TASSEL and GAPIT are compared for Width while none for DTF. Only one SNP was selected by PentaPen for the width phenotype whose corresponding gene, **AT1G65300**, could potentially be responsible for plant growth and development (Waqas et al. [2020]). For the remaining continuous phenotype (Table 4.7) of the F1 hybrids data set, PentaPen had one shared SNP (**Chr1_1327977**) with GAPIT but none with

TASSEL. The gene **AT2G28305** could potentially promote flowering which represents the function of phenotype DTF (Xiang et al. [2022]).

4.1.3 Categorical Phenotype

The accuracy was used as the performance metric for Germination Days. The accuracy values are analyzed first to study the larger value and then look for the smaller gap in train and test values to avoid over-fitting. From Figure 4.3, the observations are discussed while the exact numbers are listed from the Tables A.1, A.2, and A.3 in the Appendix.

It was observed from Tables A.1 and A.2 in Appendix or Figures 4.3a and 4.3b that Elastic net and LASSO have less over-fitting than Ridge, Group LASSO, and SGL as the gap between train and test sets using these three is wider (ranging from 0.17 to 0.27). However, Ridge has a smaller gap than Group LASSO and SGL. The accuracy of PentaPen for the test and train set is higher (0.772 and 0.741 respectively). Although the values are less than Ridge, LASSO, and Elastic net using all SNPs, the smaller gap (0.031) shows reduced over-fitting. These accuracy values of the workflow are higher than Group LASSO and SGL's values, with a smaller gap between the train and test set. Hence, PentaPen utilized the strengths of the five methods and showed high accuracy values while a smaller gap, reducing over-fitting.

Ridge accounts for **74606** SNPs out of all the SNPs before SNP pooling. LASSO, Elastic net, and PentaPen record similar SNPs (**2, 16, & 5**) as continuous phenotype, Width.

For the categorical phenotype, germination days (Table 4.8), PentaPen

had no shared SNPs and shared genes with GAPIT and TASSEL (Tables 4.3 and 4.4). It showed no shared SNPs or genes when TASSEL and GAPIT are compared. Although the transcription factor of gene **AT1G65300** identified by PentaPen could potentially mediate seed germination.

4.1.4 Evaluation of Group LASSO and SGL

For binary phenotype (Anthocyanin), similar to the analysis in Figures 4.1a and 4.1b (Tables A.1 and A.2 in the Appendix), among Group LASSO and SGL (from Figure 4.1c or Table A.3 in the Appendix), Group LASSO is preferred for the classification of predictors with higher Precision, Recall, F1-score, and AUC values, and the smaller gap in their train and test set. Using the test AUC values, it was noted that Group LASSO and SGL using filtered SNPs were better classifiers than those using all SNPs. Hence, including these two models using filtered SNPs in the proposed workflow, improves the classification accuracy of PentaPen and identifies a reasonable number of SNPs due to a reduced number of false negatives. Even when Group LASSO and SGL use a lesser number of inputs (or SNPs), still Group LASSO is noted to perform similarly to LASSO and Elastic net whereas SGL performs similarly to Ridge.

For continuous phenotypes (Width and DTF), similar to the analysis in Figures 4.2a and 4.2b (Tables A.1 and A.2 in Appendix) among Group LASSO and SGL (from Figure 4.2c or Table A.3 in Appendix), Group LASSO is preferred over SGL because of its higher and smaller R-squared and RMSE values, respectively, and a small gap between the train and test set for both the phenotypes. It is evident that Group LASSO and SGL using filtered

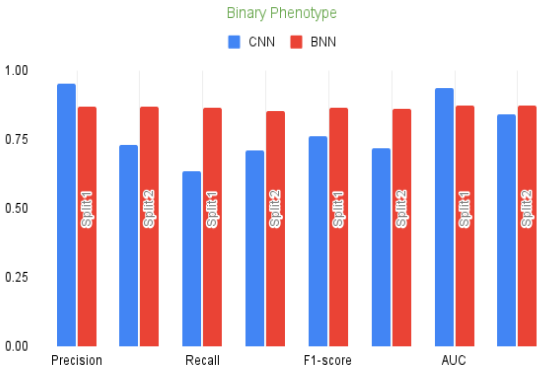
SNPs (0.403 & 0.867) reduces the gap between the train and test R-squared values more than those using all SNPs (0.058 & 0.155). Including these models reduces the chances of over-fitting by PentaPen.

For categorical phenotype (Germination days), from Table A.3 in Appendix or Figure 4.3c Group LASSO outperforms SGL with higher values (0.88 and 0.87 for train and test respectively) of Accuracy with a smaller difference (0.011). The gap is reduced for Group LASSO and SGL using filtered SNPs than those using all SNPs. Even using lesser features (or SNPs), Group LASSO and SGL still perform similarly to Ridge, LASSO, and Elastic Net.

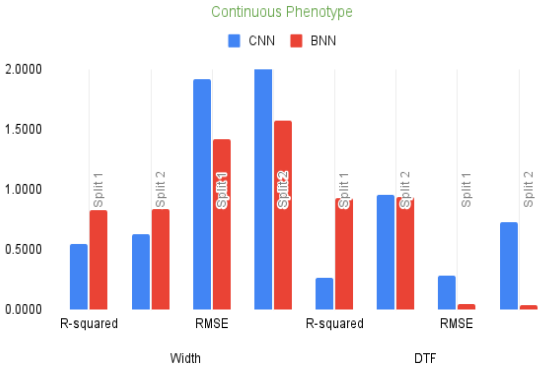
4.2 Deep Learning for GWAS

The results of the deep learning research work are presented in the following subsections, which are organized by different types of phenotypes.

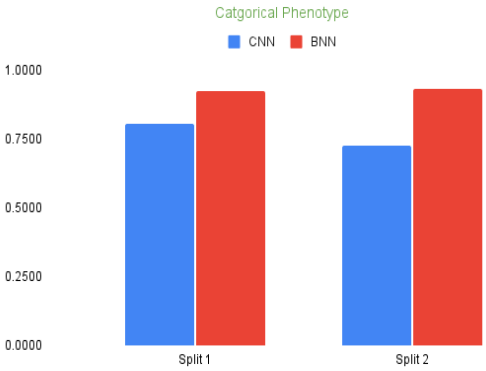
The study compares the predictive performance of CNN with BayesDL and assess their test metrics presented in Figure 4.4 (refer to Table B.1 in Appendix B for exact values). In this study, both NNs are trained using a 50% train set, and a 50% test set is used for prediction to record performance metrics. Later, the role of the train and test sets is swapped and new test performance metrics are recorded. This results in two test splits- Split 1 and Split 2 - while reporting the findings. The above results aid in assessing the superiority of BayesDL over the existing model. A few more findings of the proposed workflow are noted to gain insights into the advantages of BayesDL.



(a) Anthocyanin



(b) Width and DTF



(c) Germ

Figure 4.4: Comparison of BayesDL with a deep learning model, CNN, for all the phenotypes. The two-split test performance metrics are recorded for comparison.

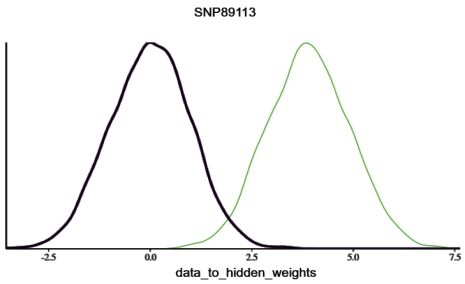
Table 4.9 displays the computation time of BayesDL, for each physical trait, observed in *A. Thaliana* which aids in noting that in spite of the extensive computational time required by Bayesian when combined with deep learning, the workflow has advantages over existing deep learning models.

Table 4.9: Computation time (in seconds) of proposed workflow based on BNN across all phenotypes

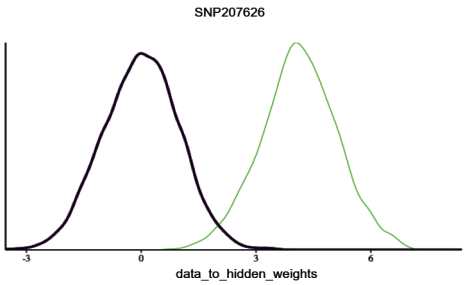
Anthocyanin	**Width**	**DTF**	**Germ**
2843	3097	3540	3384

The preliminary posterior samples check was aimed to make comparisons between the observed prior and the posterior draws from BayesDL. The null hypothesis is that apriori the SNPs (or features) are insignificant with their corresponding weights. The result for Anthocyanin, Width, DTF, and Germination Days is displayed in Figure 4.5. These findings aid in concluding that the SNPs are identified with more confidence as the prior resulted in an important posterior of the SNP.

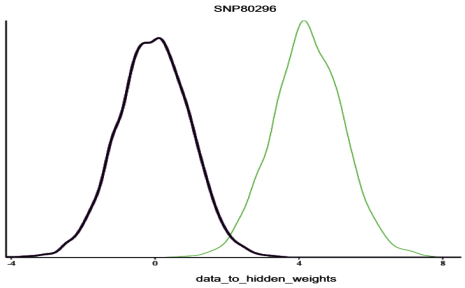
The top 10 SNPs output from BayesDL is validated to locate corresponding genes. Similar to the validation done in the penalized-based study, it was done by locating the chromosome base pair position of the particular SNP and then its corresponding gene using the Gene Model from the TAIR website. Tables 4.10, 4.11, 4.12, and 4.13 display top 10 SNPs for each phenotype and their Chromosome base pair positions, corresponding gene, and gene function. These are used to find shared SNPs and genes with GAPIT and TASSEL (Tables 4.3 and 4.4).



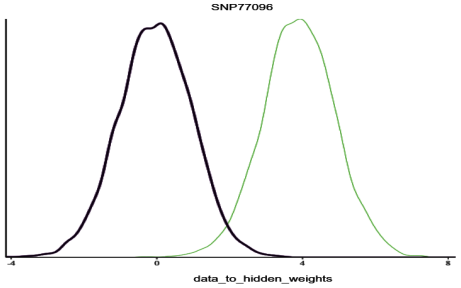
(a) 1^{st} SNP for Anthocyanin



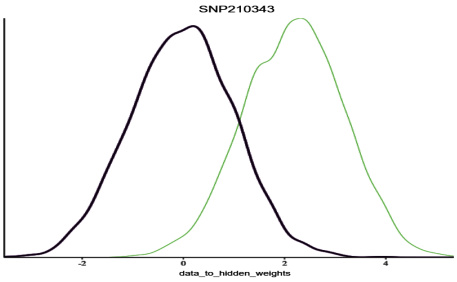
(b) 2^{nd} SNP for Anthocyanin



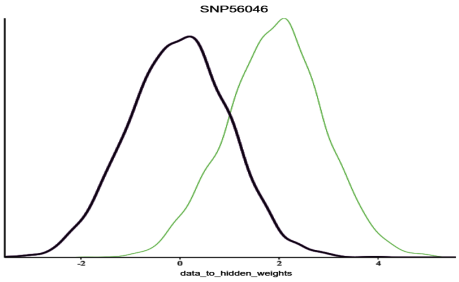
(c) 1^{st} SNP for Germ



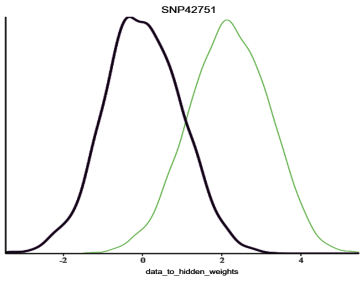
(d) 2^{nd} SNP for Germ



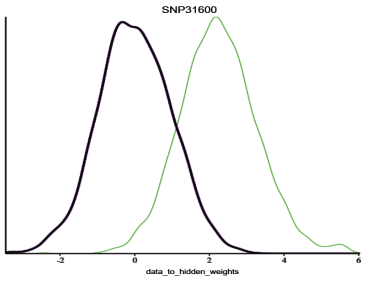
(e) 1^{st} SNP for Width



(f) 2^{nd} SNP for Width



(g) 1^{st} SNP for DTF



(h) 2^{nd} SNP for DTF

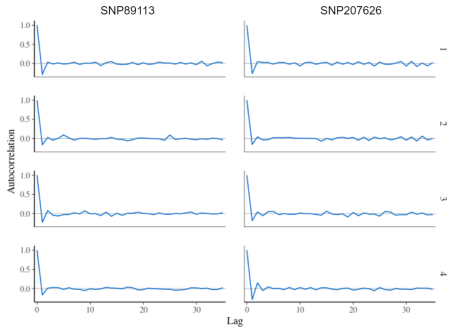
Figure 4.5: Plots for prior and posterior samples. The green line indicates the prior distribution while the purple line refers to the distribution of posterior samples.

4.2.1 MCMC Diagnostics

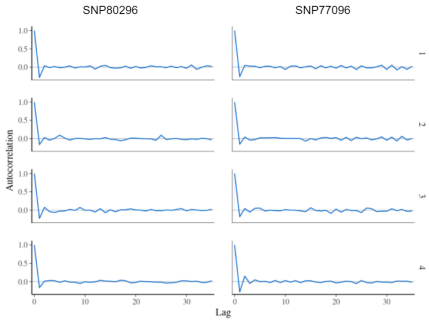
To determine the proposal values, the NUTS (an adaptive form of Metropolis-Hastings) algorithm was run for a total of 2000 iterations where the first 1000 iterations were used as burn-in. Four different chains were generated for the 2000 iterations where values from the first 1000 iterations were thrown away. Three plots were plotted for all the selected phenotypes using the weights as functions corresponding to the top 2 selected predictors (SNPs). Figures 4.6, 4.7, and 4.8 display the output of all the selected phenotypes.

Using RStan, the auto-correlation function (ACF) plots for each phenotype for all chains separately and each chain 35-lags apart can be observed in Figure 4.6. It can be observed that the generated MCMC iterations are reasonably correlated for Anthocyanin, Germination days, and Width with certain small dips few lags apart when considering each chain separately. This indicates that all four chains are highly convergent for these three phenotypes displaying the SNPs as significant features to be included for further analysis. While from Figure 4.6d it is evident that all the chains are positively convergent and all chains reached stationarity for SNP42751. However, for SNP31600, chains $1, 2, \&4$ reach stationarity within 10-lags apart whereas chain 3 reaches stationarity towards the end of the lags after some gradual geometrical decline.

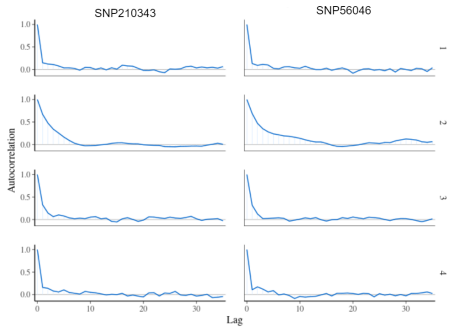
The trace plots of the generated parameters were recorded to check whether the Markov Chains reached stationarity to show the parameters' convergence to an optimal value. The convergence of the chains for the top 2 SNPs of all phenotypes was verified using RStan. The trace plots after the 1000 iterations or after the burn-in was documented. Figure 4.7 shows that



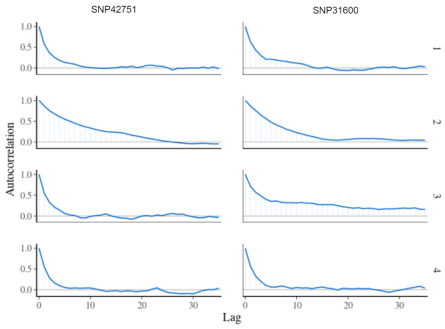
(a) Anthocyanin



(b) Germination Days



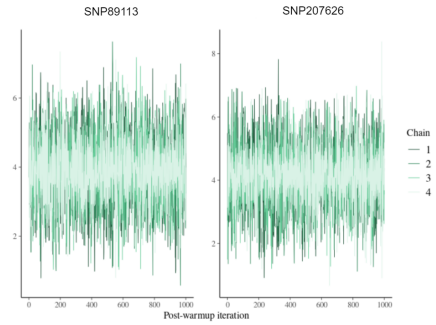
(c) Width



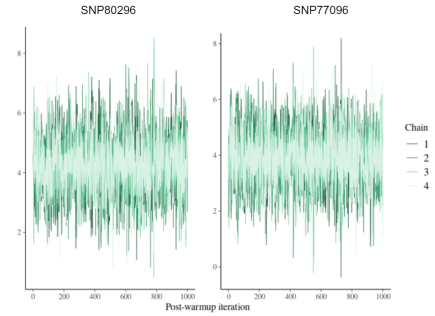
(d) DTF

Figure 4.6: Auto-correlation functions using RStan against MCMC iterations. Functions of weights corresponding to the top 2 predictors (or SNPs) for all the phenotypes. The x-axis and y-axis represent Lags and Auto-correlation values respectively.

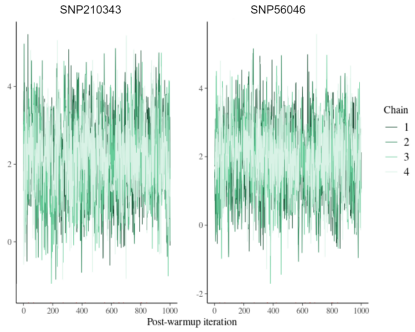
the stationarity for MCMC iterations was achieved for both the parameters (or SNPs) in Anthocyanin, Width, DTF, and Germination days. For all the phenotypes, all the chains are mixed well and contribute to stationarity. However, for DTF (see Figure 4.7d), the chain gets stuck at the 200 and 900 iterations. It is suggested to introduce more iterations and thinning in the MCMC model to improve the convergence.



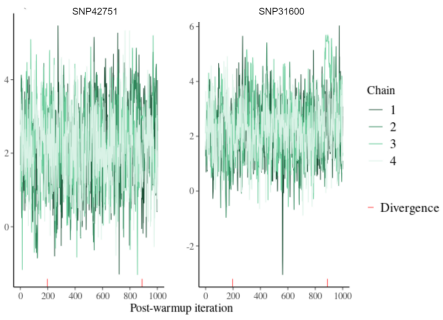
(a) Anthocyanin



(b) Germination Days



(c) Width

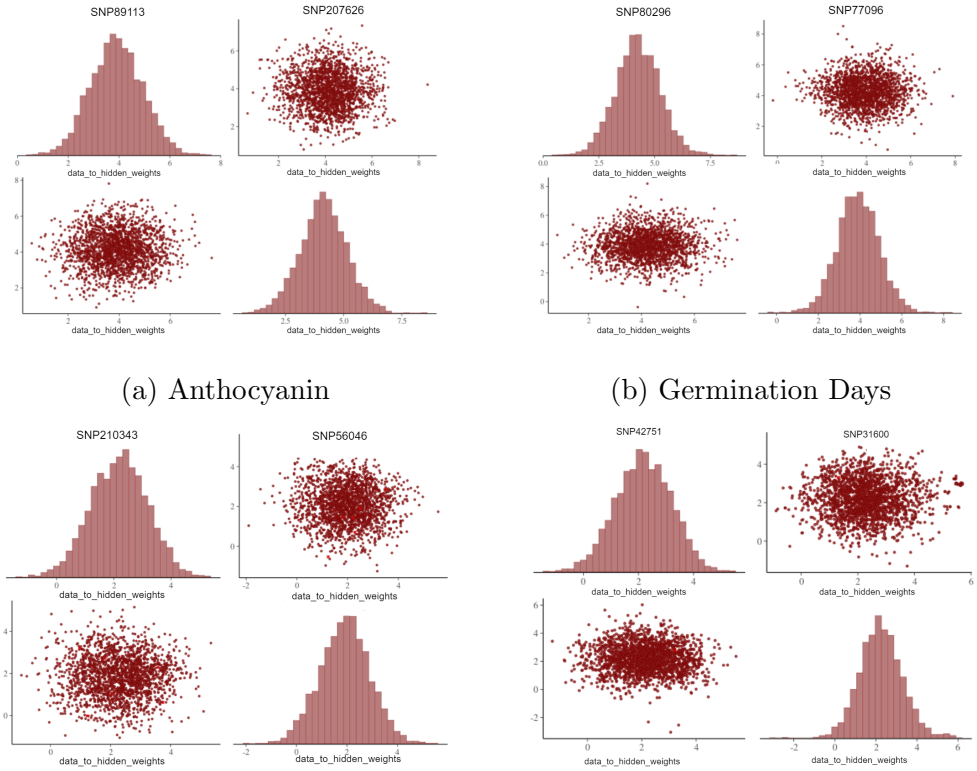


(d) DTF

Figure 4.7: Trace plots using RStan against post-warmup MCMC iterations. Here are the functions for the top 2 predictors (or SNPs) across all phenotypes and the y-axis represents the value of weights corresponding to each SNP. The trace plots assess the convergence, stability, and distribution of the weights throughout the MCMC sampling process.

This study was dealing with RStan hence, histograms in Figure 4.8 display the distributions for the features (or SNPs) using hyper-parameter values. This aids in making a reasonable comparison of whether the model gives the approximate posterior distributions. The parameters for Anthocyanin, Germination days, Width, and DTF have close enough symmetrical distributions. If the SNPs have a weight value of 0 and they lie in the low-density

region, then those SNPs are more important. In fact, the scatter plots in Figure 4.8 show that these parameters are not correlated and were randomly chosen for all the phenotypes.



(a) Anthocyanin

(b) Germination Days

(c) Width

(d) DTF

Figure 4.8: Posterior uni-variate distributions along the diagonal and bi-variate distributions along the off-diagonal using RStan against MCMC iterations. These functions represent the top 2 predictors (or SNPs) for each phenotype. The x-axis for each posterior distribution is the value of weights corresponding to the input. Whereas the scatter plot helps to check the correlation between the SNPs.

The potential scale reduction factor (PSRF) is computed by a popular statistic, Rhat. This compares between and within-chain estimates to con-

firm the convergence. The R-hat values less than 1.05 indicate that the chains mix well (Luo and Jiao [2018]). The R-hat values for the selected SNPs of Anthocyanin, Germination days, Width, and DTF are 1, 1, 1.01, and 1.01 respectively. Looking at these values, it can be concluded that the model convergence was reached for all the phenotypes.

Finally, the MCMC diagnosis indicates that important SNPs were identified for all phenotypes.

4.2.2 Binary Phenotype

Figure 4.4a or Table B.1 in Appendix shows the two test evaluation metrics for making comparisons of deep learning methodologies (NNs). The larger the value of the metric, the better the model performs for the phenotype. Reducing the difference between testing values indicates that the model is avoiding over-fitting. For Anthocyanin, it is clearly evident that BNN outperforms CNN with high values of Precision (**0.87 and 0.8708**), Recall (**0.8656 and 0.8531**), F1-score (**0.8677 and 0.8607**), and AUC (**0.8749**); BNN has a lesser gap in both splits. A high recall score indicates less number of False negatives which is an advantage of using BayesDL over CNN.

It can be noted from Figures 4.5a and 4.5b that both SNPs from Anthocyanin's BayesDL results form analogous symmetric distributions with a large shift from the standard normal distribution (the prior), depicting that both the SNPs (or features) are important. This concludes that data provided evidence against the null hypothesis and aposteriori the SNPs (or features) are important for binary phenotypes.

Table 4.10 shows the selected SNPs output from BayesDL for Anthocyanin. This binary phenotype showed **Chr4_1503923** and **AT4G03415** as a shared SNP and gene respectively with GAPIT which is evident from Tables 4.3 and 4.5. However, there were no shared SNPs or genes with TASSEL (see Tables 4.4 and 4.10). The genes **AT5G57670** (Luo et al. [2017]), **AT4G03415**, and **AT4G04920** (Iorizzo et al. [2019]) regulate the production of Anthocyanin in *Arabidopsis thaliana*.

4.2.3 Continuous Phenotype

R-squared and RMSE are used for comparing the performance of the regression NNs; their values are recorded in Figure 4.4b or Table B.1 in Appendix. The higher and lower values of R-squared and RMSE respectively, the better the performance of the NN model. A wider gap between split values increases the risk of over-fitting. For both phenotypes, it is noticed that BayesDL outperforms CNN. For instance, looking at Width values, R-squared is higher for BayesDL (**0.8313 and 0.836**) than for CNN (0.5503 and 0.6305) with a lesser gap (**0.0047**) in both the splits. The RMSE values are lower in BNN (**1.4254 and 1.5794**) than those recorded for CNN (1.9195 and 2.0317) with CNN RMSE values' gap (0.1127) larger than that for BayesDL (**0.154**).

Figures 4.5e, 4.5f, 4.5g, and 4.5h demonstrate that BayesDL results for Width and DTF have symmetric distributions that deviate significantly from the standard normal distribution, indicating that both SNPs (features) are important. These findings provide evidence against the null hypothesis and support the significance of the SNPs (features) for all continuous phenotypes.

Tables 4.3, 4.11, and 4.12 display no shared SNPs or genes for any of

the two continuous phenotypes. A similar result was noted from Tables 4.4, 4.11, and 4.12 when compared with TASSEL. This is a possibility as GAPIT and TASSEL use GLM whereas BayesDL uses probabilistic models having various distributions for SNP identification. The results of the BNN model for Width and DTF are displayed in Tables 4.11 and 4.12 respectively. The genes **AT1G51120** and **AT4G09350** are responsible for the continuous development of leaf width (Franco-Zorrilla et al. [2014]). The most relevant gene for DTF is **AT1G59940** which is responsible for flowering in the species (Hwang et al. [2002]).

4.2.4 Categorical Phenotype

Figure 4.4c or Table B.1 in Appendix records the two test accuracy scores to make comparisons of NN methods for germination days. The higher the accuracy, the better the performance of the model. For the categorical phenotype, the BayesDL exhibits superior performance compared to the CNN, achieving two split accuracy scores of **0.9241 and 0.9334**.

Figures 4.5c and 4.5d display that both SNPs from Germination days' BayesDL results have symmetric distributions with a large shift from the standard normal distribution. This finding provides evidence against the null hypothesis, concluding that the SNPs are important features.

Table 4.13 displays the results of categorical phenotype. It neither showed a shared SNP nor a shared gene with GAPIT and TASSEL (see Tables 4.3 and 4.4). However, this phenotype contributes to the germination of the plant and a few genes (**AT1G03890, AT3G31950, and AT4G35950**) were found responsible for it (Nagano et al. [2001], Maruta et al. [2021]).

Table 4.5: SNP Validation for Anthocyanin. The table displays the top 10 SNPs reported by PentaPen. Bold values of genes and SNPs have the true function of the phenotype.

SNPs	**Chr_bp**	**Gene**	**Gene Function**
SNP41810	Chr1_24362213	AT1G65540	LETM1-like protein. No function found. TAIR.
SNP18139	Chr1_10547614	AT1G30070	SGS domain-containing protein. No function found. TAIR.
SNP25069	Chr1_14026469	AT1G36980	transmembrane 50A-like protein. No function found. NCBI.
SNP81693	Chr3_725078	AT3G03140	histone binding, protein binding. NCBI.
SNP34101	**Chr1_21234054**	**AT1G56650**	Production of Anthocyanin Pigment. TAIR.
SNP35159	Chr1_21692775	AT1G58390	Disease resistance protein (CC-NBS-LRR class) family. ADP binding. TAIR.
SNP36875	Chr1_22492447	AT1G61070	Predicted to encode a PR (pathogenesis-related) protein. TAIR.
SNP26674	Chr1_16547538	No gene	–
SNP85074	Chr3_2738646	AT3G08970	Can compensate for the growth defect. Also shows similarity to HSP40 proteins and is induced by heat stress. NCBI.
SNP45199	Chr1_26199451	No gene	–

Table 4.6: SNP Validation for Width. The table displays the top 10 SNPs reported by PentaPen. The SNPs and genes which are bold, show the true function of Width.

SNPs	**Chr_bp**	**Gene**	**Gene Function**
SNP51671	Chr1_30250888	No gene	–
SNP41509	**Chr1_24255390**	**AT1G65300**	DNA-binding transcription factor activity. Gene Ontology.
SNP18948	Chr1_11016927	No gene	–
SNP98767	Chr3_9842540	No gene	–

Table 4.7: SNP Validation for DTF. The table displays the top 10 SNPs reported by PentaPen. The highlighted genes and SNPs show the true function of DTF.

SNPs	**Chr_bp**	**Gene**	**Gene Function**
SNP47857	Chr1_25311984	AT1G67540	Protein Associated with Lipid Droplets. Gene Ontology.
SNP11327	Chr1_7300804	AT1G20950	Phosphofructokinase family protein. ATP binding. NCBI.
SNP10090	Chr1_6608183	AT1G19120	mRNA. No function found. TAIR.
SNP76193	Chr2_12721604	AT2G29790	Encodes a Maternally expressed gene (MEG) family protein. NCBI.
SNP30293	Chr1_19149817	AT1G51640	enables phosphatidylinositol-4,5-bisphosphate binding. TAIR.
SNP58484	Chr2_835205	No gene	–
SNP74899	**Chr2_12082004**	**AT2G28305**	enables hydrolase activity. TAIR.
SNP2195	Chr1_1219863	AT1G04490	hypothetical protein. No function found. TAIR.
SNP102004	Chr3_9590645	AT3G26200	mRNA. iron ion binding. NCBI.
SNP130361	Chr4_5198089	AT4G08250	DNA-binding transcription factor activity. TAIR.

Table 4.8: SNP Validation for Germination Days. The table displays the potentially important SNPs reported by PentaPen. The gene and SNPs showing the true function of the phenotype are highlighted.

SNPs	**Chr_bp**	**Gene**	**Gene Function**
SNP51671	Chr1_30250888	No gene	–
SNP41509	**Chr1_24255390**	**AT1G65300**	DNA-binding transcription factor activity. Gene Ontology.
SNP13281	Chr1_7797023	AT1G22090	Protein of unknown function. Gene Ontology.
SNP34187	Chr1_21325557	AT1G57580	F-box family protein of unknown function. NCBI.
SNP62673	Chr2_7129923	AT2G16440	Protein binding and DNA replication. NCBI.

Table 4.10: SNP Validation for Anthocyanin. The table displays the top 10 SNPs reported by BayesDL. The highlighted genes and SNPs are found to show characteristics of Anthocyanin.

SNPs	**Chr_bp**	**Gene**	**Gene Function**
SNP89113	Chr3_5158241	No gene	–
SNP207626	**Chr5_23362168**	**AT5G57670**	Enables protein serine/threonine kinase activity. TAIR.
SNP27368	Chr1_16970522	AT1G44900	A protein essential to embryo development. Overexpression results in altered root meristem function. NCBI.
SNP129572	**Chr4_2500993**	**AT4G04920**	Enables transcription coregulator activity. TAIR.
SNP161227	Chr5_205246	AT5G01510	Protein of unknown function. NCBI.
SNP25224	Chr1_14138318	AT1G37113	Hypothetical protein. No function found. NCBI.
SNP197173	Chr5_18231342	AT5G45110	NPR1-like protein 3. Enables identical protein binding, protein binding, and salicylic acid binding. TAIR.
SNP89096	Chr3_5144422	AT3G15280	No function found. NCBI.
SNP27389	Chr1_16980256	AT1G44910	Enables RNA binding, RNA polymerase binding. NCBI.
SNP127389	**Chr4_1503923**	**AT4G03415**	Protein phosphatase interacts with AGB1 and is localized to the plasma membrane. TAIR.

Table 4.11: SNP Validation for Width. The table displays the top 10 SNPs reported by BayesDL. The bold values of SNPs and genes are responsible for the observed trait.

SNPs	**Chr_bp**	**Gene**	**Gene Function**
SNP210343	Chr5_24959340	No gene	–
SNP56046	Chr2_2309063	AT2G05970	F-box family protein with a domain of the unknown function (DUF295). TAIR.
SNP52086	Chr2_100349	No gene	–
SNP30943	**Chr1_18938157**	**AT1G51120**	Enables DNA-binding transcription factor activity. TAIR.
SNP134906	**Chr4_5931715**	**AT4G09350**	Chaperone DnaJ-domain superfamily protein. No function found. NCBI.
SNP57073	Chr2_2994278	AT2G07212	Transposable element gene. No function found. NCBI.
SNP51689	Chr1_30259387	AT1G80480	Plastid transcriptionally active 17. No function found. TAIR.
SNP178757	Chr5_9128487	AT5G26130	CAP (Cysteine-rich secretory proteins). No function found. NCBI.
SNP144027	Chr4_9309784	No gene	–
SNP100922	Chr3_10838525	AT3G28840	Hypothetical protein (DUF1216). No function found. TAIR.

Table 4.12: SNP Validation for DTF. The table displays the top 10 SNPs reported by BayesDL. There is only one gene or SNP that is associated with DTF, which is highlighted in the table.

SNPs	**Chr_bp**	**Gene**	**Gene Function**
SNP42751	Chr1_23756683	No gene	–
SNP31600	Chr1_19709266	AT1G52910	No function found. TAIR.
SNP96314	Chr3_6498906	No gene	–
SNP198764	Chr5_22834746	No gene	–
SNP51715	Chr1_27340888	No gene	–
SNP57031	Chr2_317144	AT2G01720	Ribophorin I. enables protein binding. NCBI.
SNP50026	Chr1_26410107	AT1G70130	Enables kinase activity, protein serine/threonine kinase activity. TAIR.
SNP194264	Chr5_20105999	AT5G49540	Rab5-interacting family protein. No function found. NCBI.
SNP38498	Chr1_22602399	No gene	–
SNP36437	**Chr1_22066825**	**AT1G59940**	phosphorelay response regulator activity, protein binding. TAIR.

Table 4.13: SNP Validation for Germination Days. The table displays the top 10 SNPs reported by BayesDL. The highlighted genes and SNPs are found to be associated with the phenotype.

SNPs	**Chr_bp**	**Gene**	**Gene Function**
SNP80296	Chr2_19554169	AT2G47700	Enables protein binding, ubiquitin-protein transferase activity. TAIR.
SNP77096	Chr2_17432976	AT2G41790	Enables metalloendopeptidase activity. NCBI.
SNP152576	Chr4_13295619	No gene	–
SNP6672	**Chr1_3990189**	**AT1G03890**	Enables nutrient reservoir activity. TAIR.
SNP189759	Chr5_15364726	AT5G38386	F-box/RNI-like superfamily protein. No function found. NCBI.
SNP7494	Chr1_4392379	AT1G12890	Enables DNA-binding transcription factor activity, protein binding. TAIR.
SNP76252	Chr2_16915416	No gene	–
SNP105241	**Chr3_12951052**	**AT3G31950**	Nucleic acid-binding/zinc ion-binding protein. TAIR.
SNP158329	**Chr4_17025614**	**AT4G35950**	GTP binding, GTPase activity, protein binding, protein kinase binding. NCBI.
SNP80122	Chr2_19469107	No gene	–

Chapter 5

Discussion and Conclusion

Two workflows are proposed to find the important SNPs. Users can identify reduced numbers of SNPs using the PentaPen (a penalized-based workflow). By using the optimal number of hidden layers and units without changing their activation functions for each run, researchers can employ BayesDL (a BNN-based workflow) to identify important SNPs. The SNPs output from both the workflows were utilized to find the shared SNPs with each other and GWAS software (GAPIT and TASSEL). Using the final top 10 SNPs from both the workflows, the SNP validation was carried out. The superiority of PentaPen and BayesDL was confirmed by comparing the performance of the proposed workflows with some of the existing models based on performance metrics. The study provides a guideline for the researchers to choose an appropriate workflow for their research with proposed workflows in terms of identified SNPs, data dimensionality, model complexity, and prior distributions (in BayesDL).

5.1 Identified SNPs

This section covers the biological interpretations and conclusions of the identified SNPs from individual penalized models, PentaPen, and BayesDL.

As reported in Table 4.1, penalized models using all SNPs report a larger number of SNPs for the binary, continuous, and categorical phenotype than PentaPen. This can be a relevant observation since Ridge and LASSO models, hence, Elastic net, are dependent on trait architecture (Seymour et al. [2016]). Ridge shrinks the features' coefficient to 0 but does not reduce the features. Due to the sparsity in SGL, the union of SNPs selected from Group LASSO and SGL gave a reduced number of important SNPs; this is more promising because PentaPen identifies SNPs by combining the essential properties of feature selection from all the five models. PentaPen helps to reduce the impact of noise and variability in the data, making PentaPen less susceptible to over-fitting. It can be concluded that PentaPen selects reasonable numbers of SNPs compared to single penalized models as they either give too many (like in Ridge) or too few (as in SGL) SNPs which can not be useful for biological interpretation. Hence, PentaPen was able to leverage the beneficial properties of the five penalized models by choosing SNPs and showing similar prediction performance when compared with the five models.

According to the results, the final set of outputs from both the proposed workflows had neither shared SNPs nor shared genes when compared with each other. This is possible due to various reasons: different statistical models, different numbers of input predictors, and model complexity. PentaPen's output did not show shared SNPs or genes with GAPIT or TASSEL

for Width (continuous phenotype) and Germination days (categorical phenotype); whereas, there was a shared gene and SNP with GAPIT but not with TASSEL for Anthocyanin (binary phenotype) and DTF (continuous phenotype) respectively. BayesDL's output for Anthocyanin (binary phenotype) showed a shared SNP and gene with GAPIT but not with TASSEL while other phenotypes did not have shared SNPs or genes with both GWAS software. However, there were SNPs and genes shared between GAPIT and TASSEL for Anthocyanin (binary phenotype) and Width (continuous phenotype) except for DTF (continuous phenotype) and Germination days (categorical phenotype). This is possible because GAPIT and TASSEL employ GLM, whereas PentaPen uses five distinct penalized approaches, and BayesDL utilizes various probabilistic models and distributions to identify the potentially important SNPs.

Hence, it is advised to locate the potentially important SNPs using both the GWAS software and our study workflows because they use different statistical methods, pre-processing steps, hyper-parameter choices, and the inclusion of prior knowledge. PentaPen, BayesDL, and both GWAS software possessed some of the genes necessary for the phenotypic traits found in *Arabidopsis thailana* species. These techniques can be applied to a phenotype of this species and are thought to be complementary to one another. Further, it is also recommended to analyze multiple instances of the same phenotype (binary, continuous, or categorical) from the workflows to increase the reliability of results for identifying shared SNPs or genes with GWAS software. The SNP validation proved that PentaPen and BayesDL were able to find a few promising SNPs which showed the function of that phenotype; this concludes that both the workflows perform similarly in terms of identified SNPs.

5.1.1 Linkage Disequilibrium

The non-random association of alleles at two or more loci is called Linkage disequilibrium (LD). The LD was not removed while pre-processing the data due to the removal of SNPs with high LD during the SNP calling step. A few identified SNPs from PentaPen and BayesDL resulted in LD with each other. For example, in DTF (continuous phenotype) Chr4_5198089 is highly correlated with Chr1_ 26410107. Since LD is measured as squared correlation, this leads to redundant information from the cluster of highly correlated SNPs. These SNPs may not contribute independently to the workflow's performance which further impacts the interpretability of the results. Despite the mentioned limitations, LD patterns reflect the genetic architecture of the studied population and inform about the genetic regions with high LD which further aids in identifying rare variants. However, it was found that among the SNPs identified by PentaPen for Anthocyanin (binary phenotype), there were only two highly correlated SNPs (Chr1_22492447 and Chr3_2738646). Similarly, for Anthocyanin, BayesDL identify two highly correlated SNPs (Chr3_5158241 and Chr5_205246). For other phenotypes, both PentaPen and BayesDL do not identify any highly correlated SNPs.

5.2 Models' Performance

This section covers the interpretation and conclusion of each penalized model, PentaPen, and BayesDL.

The rigorous comparison of penalized models using all SNPs depicts that

Ridge outperforms Group LASSO and SGL whereas LASSO and Elastic net should be preferred over Ridge, Group LASSO, and SGL for classifying/predicting the groups/responses accurately. LASSO and Elastic Net demonstrate the efficacy of a good classification/regression model evident from the findings across all the phenotypes. Okser et al. [2014] also showed that LASSO and Elastic Net have similar prediction behavior for two whole-genome SNP data. However, Romagnoni et al. [2019] found that Ridge, LASSO, and Elastic Net provide similar results with optimized evaluation metrics on the Immunochip data set. For all the phenotypes, the test scores of Group LASSO and SGL using all SNPs indicate that they perform worse than Ridge. Including Ridge in the union for the input set of Group LASSO and SGL using filtered SNPs provides a larger set of input variables which helps reduce the risk of over-fitting and improve the model's performance.

Between the results of Group LASSO and SGL using filtered SNPs, the former generally perform better, as shown in Table A.3 in Appendix or Figure 4.3. But SGL using filtered SNPs has group-wise and within-the-group sparsity (Simon et al. [2013]) and can potentially benefit PentaPen. SGL was discovered to provide comparatively lower testing metrics value than Group LASSO. Instead, Group LASSO using filtered SNPs offered less variance between the values of the training and testing parameters. It was also noticed that even if Group LASSO and SGL use a lesser number of inputs (or SNPs) when trained to get the final output of PentaPen than other penalized methods using all SNPs, they perform equally well as Ridge, LASSO, and Elastic net & even better than Group LASSO and SGL using all SNPs. This finding leads us to the conclusion that PentaPen uses models that reduce the bias and variance of their models, and obtain more stable estimates of the coefficients.

It was found that PentaPen performs similarly to Ridge, LASSO, and Elastic Net based on its evaluation metrics while producing reduced numbers of SNPs for the binary phenotype. For continuous and categorical phenotypes, PentaPen still performs similarly to LASSO and Elastic Net. However, for binary phenotype, PentaPen outperforms Group LASSO and SGL using all SNPs while reducing over-fitting evident in Group LASSO and SGL. For continuous and categorical phenotypes, PentaPen outperforms Ridge along with Group LASSO and SGL using all SNPs. This further guarantees that PentaPen could combine the strengths of five models to make better predictions and identify SNPs with confidence. Hence, based on the performance metrics calculated in this study, it can be concluded that PentaPen is superior to some of the existing methodologies with the advantages of combined strengths of the models used.

A thorough comparison of deep learning approaches depicts that the BNN model should be chosen above CNN, for both classification and regression. Similar superiority of BNNs over traditional models was noted by Beam et al. [2014]. Bayesian combined with deep learning is advantageous in various scenarios because they provide a framework for probabilistic modeling that allows for uncertainty quantification. BayesDL's averaging of samples from the posterior distribution reduces variability and over-fitting, unlike traditional deep learning models that output a single prediction. This property enhances generalization to new data, making BayesDL more reliable than traditional models in limited sample size instances (Gal and Ghahramani [2016], Blundell et al. [2015]). It can also be concluded that with fewer samples in phenotypes, the BNN model was advantageous over CNN. The possible reasons for the better performance of BayesDL are that it is flexible in handling sparse data with a small sample size; in contrast, CNN requires

complete data (based on the data used in this study, the rows need to be replicated to make it a full data) to handle such data (Deist et al. [2018]). Additionally, BNNs incorporate prior knowledge or constraints, which can be helpful in biological applications where some knowledge about the data is available (Liu et al. [2019]). The use of prior information about the distribution of the parameters help to regularize the model, which can help prevent over-fitting and improve the performance of the model (Ghahramani [2015]). Hence, in conclusion, based on the performance metrics calculated in this study, the BayesDL is superior to CNN with advantages of the properties of Bayesian.

5.2.1 Controlling R-squared Errors: Impact on Performance

Controlling R-squared errors during evaluation helps in gaining more reliable estimates of workflows' performance. There are various ways to control R-squared errors in PentaPen and BayesDL. This can be done through cross-validation by assessing the average performance of the workflows and mitigating the influence of over-fitting. Although cross-validation was utilized in PentaPen, this technique can be used in BayesDL to assess the impact. Regularization methods are also effective in controlling over-fitting and R-squared errors. In PentaPen, the penalty terms can be tuned in individual models while in BayesDL, optimized prior variance can be used. This will result in finding a balance between model performance and generalizability. Further, controlling R-squared errors might enable unbiased comparison of different models, prevents over-fitting, and generalize better to unseen data.

5.3 Data Dimensionality and Model Complexity

Table 5.1 displays the data dimensionality and model complexity of PentaPen and BayesDL (Li et al. [2018b]).

Table 5.1: Comparison of two developed workflows based on the data dimensionality and model complexity. Here, n, p, and o denote the number of samples, predictors, and output dimensions or classes of the data set respectively. The number of folds, iterations, chains, hidden layers, and nodes in each layer is represented by k, i, c, m, and N respectively. p_{pool} are the number of SNPs from SNP Pooling. The number of SNPs preliminary selected from the hypothesis test are given by p_{filter}. Lastly, P is the total number of parameters of the workflow.

Workflows	**Data's dimension**	**Model's dimension**	**Total parameters**	**# of iterations**	**Time complexity**
PentaPen	n× p	n× p	$3*p+2*p_{pool}$	5	O($k*p^2*n*i+n^2$)
BayesDL	n× p_{filter}	m× N	$p+100*p+50*o$	1000 (post warmup)	O($i*c*P*n*p_{filter}$)

Based on the data dimensionality, the study findings suggest various options for researchers who are dealing with whole-genome SNP data. Table

5.1 displays that the workflows have different data dimensionality and interested researchers can choose a workflow based on the data they are using. Researchers using all SNPs (or features) as the input may prefer using PentaPen. By simply inputting their data and pre-processing it according to the variability in the nucleotide notations in the data, researchers can obtain the final set of SNPs using PentaPen. However, BayesDL requires significant computational resources (32GB computation RAM) even after preliminary feature selection. Therefore, if researchers have SNPs in the range of 90 to 250, the suggested approach for SNP identification would be to use BayesDL. However, if they have a smaller sample size than 175, then after selecting SNPs less than the number of samples (p<n), using BayesDL would be recommended due to its advantage of having prior knowledge about the parameters, reducing over-fitting, and increasing the prediction performance.

Based on the model complexity, the research has different suggestions for choosing the workflow for SNP identification. It can be noticed from Table 5.1 that comparison based on the time complexity of the workflows is challenging as they depend on different parameters. However, it is evident that BayesDL has model dimensions based on the number of hidden layers and nodes in each layer, which makes the model more complex than PentaPen. Additionally, the number of parameters and iterations of BayesDL are more than those used in PentaPen. This leads to the conclusion that BayesDL is more complex than PentaPen and it may take more computational time.

Additionally, the time complexity of PentaPen was compared with the penalized models. The time complexity of Ridge, LASSO, and Elastic Net is O($k*p^2*n*i$). The time complexity for PentaPen is O($k*p^2*n*i+n^2$). Here, k represents the number of folds, p denotes the number of input predictors or

SNPs, n is the number of input samples, and i gives the number of iterations used to train and validate the workflow. It is noticed that the time complexity of PentaPen is greater than that of individual penalized models due to the Hierarchical clustering involved in group formation.

Furthermore, the computation time of PentaPen with each penalized model was compared. Ridge, LASSO, and Elastic net used all SNPs as input, and the computation was more complex in Ridge than in the other two models as seen from Table 4.2. This is because the tuning parameter λ in Ridge takes a long time when dealing with small-p(SNP)-large-n(sample) data. Despite taking longer to compute than the Elastic net and LASSO, PentaPen has the advantage of combining properties of five penalized models to identify more important SNPs than any of these five models. PentaPen utilizes parallel computing to speed up the computation of multiple penalized models. This workflow takes the union of selected SNPs from Ridge, LASSO, and Elastic net as input for Group LASSO and SGL; Group LASSO and SGL further output the important SNPs utilizing their strongly group-sparse property (Huang and Zhang [2010]). This can be noticed from Table 4.1 that PentaPen finally produces reduced numbers of the most important SNPs. These SNPs were further validated to find corresponding genes' properties with the phenotype.

The utilization of probabilistic models that are integrated with the data has been observed to enhance the prediction power while using BNN for prediction (Neal and Zhang [2006]). Despite the high computational time (displayed in Table 4.9) and computational memory requirements (a server of at least 32GB RAM) of BayesDL, it offers several advantages. It results in high prediction power, mitigates the issue of over-fitting, and can effectively

handle data sets with fewer sample sizes.

5.4 Prior Distributions

The researchers who are choosing to use BayesDL may find it challenging to obtain prior distributions or beliefs of the genome data or parameters. However, they may refer to the existing literature or the Bayesian statisticians to gain some insight into the expected ranges of values for the parameters. In some cases, domain-specific knowledge can also provide valuable information for setting priors. Researchers may also choose to use non-informative priors or priors that are not affected by the choice of hyperparameters to reduce the impact of prior specifications on the final results. Overall, while it may not always be feasible to obtain prior distribution for all parameters, using uninformative priors can still improve the chances of reliability on BayesDL.

5.5 Limitations and Guidelines for BayesDL

Based on the experimental results and discussion in the above sections, there are several challenges to running BayesDL on whole-genome data. BayesDL involves sampling from posterior distributions that require storing multiple weight samples. It is computationally expensive and consumes more memory, especially for high-dimensional data and complex models. The large number of weight samples also leads to slow convergence to the true posterior. In conclusion, scaling BayesDL to large datasets can be challenging due to the increased computational and memory requirements.

Although BayesDL is recommended for smaller sets of SNPs (90-250 SNPs) and poses several challenges, the advantages of Bayesian methods in handling uncertainty, regularization, prior knowledge, and reducing overfitting make it a promising approach. BayesDL can be used for whole-genome SNP data with hundreds of thousands of SNPs by addressing the challenges in several ways. Instead of running BayesDL on the entire dataset at once, use mini-batch sampling to process smaller subsets of data (Kuhn et al. [2020]). BayesDL can be parallelized to leverage multiple processors or GPU resources. The traditional MCMC methods for Bayesian inference may be computationally intensive, but advanced techniques like Stochastic Gradient Langevin Dynamics (Welling and Teh [2011]) can be employed to scale to whole-genome data. Further, BayesDL can be trained in a progressive learning technique, starting with a small subset of the data, and gradually incorporating more data in subsequent iterations. These methods help reduce memory requirements and enable the handling of larger datasets.

5.6 Contributions

The thesis has made significant contributions that can be summarized as follows:

- The thesis provided a rigorous comparison of penalized models for interested researchers about the best available methods for feature selection.
- The thesis also provided a guideline for the bioinformatics research community by comparing the performance of deep learning models.
- In the thesis, two new workflows have been proposed. PentaPen com-

bines various penalized methodologies, while BayesDL combines the Bayesian methods with deep learning. Both of these workflows have demonstrated superior performance metrics compared to some of the top-performing models. Additionally, these workflows have identified SNPs with greater confidence by reducing false negatives, highlighting their potential as powerful tools in SNP identification.

- BayesDL is user-friendly as it utilizes only one model whose hyper-parameters can be easily tuned in Stan. BayesDL does not require extensive knowledge in preliminary feature selection and training the model in whole-genome SNP data, yet it has identified a highly relevant set of differentially expressed SNPs.

5.7 Future work

This research has several potential works for the future, which are listed below:

- To expand the current research, a study of how the model differences between GWAS, penalized, and deep learning models influence the varied SNPs identified from them could be included.
- Another potential work is to combine PentaPen with different GWAS software to get the important SNPs with high confidence.
- More work can be done for selecting input SNPs of deep learning models using additional algorithms and related R packages, such as MXM developed by Tsagris and Tsamardinos [2019], which also implements

statistical and conditional independence tests (like Fisher and Spearman correlation).

- An expanded approach for BayesDL is to run the model for 5 iterations to compute averaged performance metrics. This may be done in parallel which results in decreased computation time and RAM and also has the potential to enhance the model's performance. It would be essential to investigate the impact of this approach on relevant SNP selection further.

- Researchers could focus on optimizing PentaPen and BayesDL for plant-specific characteristics (such as genomic variations and polyploidy) by developing specific pre-processing techniques like array customization of samples (Sun et al. [2020]). They may integrate domain-specific knowledge to enhance SNP identification and interpretation in diverse plant genomes. Collaboration with plant geneticists can help in generalizing the workflows for diverse plant genomes and foster advancements in plant genomics research.

- The issue being investigated has the potential to be rephrased as a multi-task prediction challenge, which could then be tackled with the MTPS R package developed by Xing et al. [2020]. This package can execute various methods, including GLM, k-nearest neighbor classification, and Support Vector Machine. The researchers have claimed that MTPS offers better prediction performance than neural networks. It is also possible to combine SNP identification with MTPS, which could lead to even more promising outcomes.

- Analyzing more instances of the phenotypes of the same data type (binary, continuous, or categorical) may increase the reliability of the

results in identifying the shared SNPs or genes with the GWAS software. The researchers may use the same or varied genotype data for this potential future work.

- The research can be expanded to address the issue of LD and reduce correlated SNPs by performing LD pruning or LD clumping or LD adjustment during pre-processing.

Bibliography

Naomi E Allen, Cathie Sudlow, Tim Peakman, Rory Collins, and Uk biobank. Uk biobank data: come and get it, 2014.

Encyclopædia Britannica. single nucleotide polymorphism. cambridge university press, the editors of encyclopædia britannica, 2014, 2019.

Arthur Korte and Ashley Farlow. The advantages and limitations of trait analysis with gwas: a review. Plant methods, 9(1):1–9, 2013. doi: 10.1186/1746-4811-9-29.

R. Bellman and R. Kalaba. On adaptive control processes. IRE Transactions on Automatic Control, 4(2):1–9, 1959. doi: 10.1109/TAC.1959.1104847.

Pei Xu, Shizhong Xu, Xiaohua Wu, Ye Tao, Baogen Wang, Sha Wang, Dehui Qin, Zhongfu Lu, and Guojing Li. Population genomic analyses from low-coverage rad-seq data: a case study on the non-model cucurbit bottle gourd. The Plant Journal, 77(3):430–442, 2014. doi: https://doi.org/10.1111/tpj.12370.

Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. ACM computing surveys (CSUR), 50(6):1–45, 2017. doi: https://doi.org/10.1145/3136625.

Bhavithry Sen Puliparambil, Jabed Tomal, and Yan Yan. Benchmarking penalized regression methods in machine learning for single cell rna sequencing data. In Comparative Genomics: 19th International Conference, RECOMB-CG 2022, La Jolla, CA, USA, May 20–21, 2022, Proceedings, pages 295–310. Springer, 2022. doi: https://doi.org/10.1007/978$-$3$-$031$-$06220$-$9_17.

M. McCombe. Intro to feature selection methods for data science, 2019.

Ioannis Tsamardinos, Giorgos Borboudakis, Pavlos Katsogridakis, Polyvios Pratikakis, and Vassilis Christophides. A greedy feature selection algorithm for big data of high dimensionality. Machine learning, 108:149–202, 2019. doi: 10.1007/s10994-018-5748-7.

Ioannis Kavakiotis, Patroklos Samaras, Alexandros Triantafyllidis, and Ioannis Vlahavas. Fifs: A data mining method for informative marker selection in high dimensional population genomic data. Computers in biology and medicine, 90:146–154, 2017. doi: 10.1016/j.compbiomed.2017.09.020.

JiaRui Li and Tao Huang. Predicting and analyzing early wake-up associated gene expressions by integrating gwas and eqtl studies. Biochimica et Biophysica Acta (BBA)-Molecular Basis of Disease, 1864(6):2241–2246, 2018. doi: 10.1016/j.bbadis.2017.10.036.

Hyelynn Jeon and Sejong Oh. Hybrid-recursive feature elimination for efficient feature selection. Applied Sciences, 10(9):3211, 2020. doi: https://doi.org/10.3390/app10093211.

Seyedali Mirjalili and Seyedali Mirjalili. Genetic algorithm. Evolutionary Algorithms and Neural Networks: Theory and Applications, pages 43–55, 2019. doi: https://doi.org/10.1007/978-3-319-93025-1_4.

James Kennedy and Russell Eberhart. Particle swarm optimization. In Proceedings of ICNN'95-international conference on neural networks, volume 4, pages 1942–1948. IEEE, 1995. doi: 10.1109/ICNN.1995.488968.

Shuangge Ma and Jian Huang. Penalized feature selection and classification in bioinformatics. Briefings in bioinformatics, 9(5):392–403, 2008. doi: https://doi.org/10.1093/bib/bbn027.

Haoyang Liu and Rina Foygel Barber. Between hard and soft thresholding: optimal iterative thresholding algorithms. Information and Inference: A Journal of the IMA, 9(4):899–933, 2020. doi: 10.1093/imaiai/iaz027.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Robert Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68(1):49–67, 2006. doi: 10.1111/j.1467-9868.2005.00532.x.

Noah Simon, Jerome Friedman, Trevor Hastie, and Robert Tibshirani. A sparse-group lasso. Journal of computational and graphical statistics, 22 (2):231–245, 2013. doi: 10.1080/10618600.2012.681250.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. Journal of the royal statistical society: series B (statistical

methodology), 67(2):301–320, 2005. doi: 10.1111/j.1467-9868.2005.00503.x.

Huijiang Gao, Yang Wu, Jiahan Li, Hongwang Li, Junya Li, and Runqing Yang. Forward lasso analysis for high-order interactions in genome-wide association study. Briefings in bioinformatics, 15(4):552–561, 2014. doi: 10.1093/bib/bbt037.

Jin Liu, Jian Huang, Shuangge Ma, and Kai Wang. Incorporating group correlations in genome-wide association studies using smoothed group lasso. Biostatistics, 14(2):205–219, 2013. doi: 10.1093/biostatistics/kxs034.

Wessel N Van Wieringen, David Kun, Regina Hampel, and Anne-Laure Boulesteix. Survival prediction using gene expression data: a review and comparison. Computational statistics & data analysis, 53(5):1590–1603, 2009. doi: 10.1016/j.csda.2008.05.021.

Hege M Bøvelstad, Ståle Nygård, Hege L Størvold, Magne Aldrin, Ørnulf Borgan, Arnoldo Frigessi, and Ole Christian Lingjærde. Predicting survival from microarray data—a comparative study. Bioinformatics, 23(16):2080–2087, 2007. doi: 10.1093/bioinformatics/btm305.

Joseph O Ogutu, Torben Schulz-Streeck, and Hans-Peter Piepho. Genomic selection using regularized linear regression models: ridge regression, lasso, elastic net and their extensions. In BMC proceedings, volume 6, pages 1–6. Springer, 2012. doi: 10.1186/1753-6561-6-s2-s10.

Vân Anh Huynh-Thu, Yvan Saeys, Louis Wehenkel, and Pierre Geurts. Statistical interpretation of machine learning-based feature importance scores for biomarker discovery. Bioinformatics, 28(13):1766–1774, 2012. doi: https://doi.org/10.1093/bioinformatics/bts238.

Ronald Fisher. The analysis of variance with various binomial transformations. Biometrics, 10(1):130–139, 1954. doi: https://doi.org/10.2307/3001667.

Tessel E Galesloot, Kristel Van Steen, Lambertus ALM Kiemeney, Luc L Janss, and Sita H Vermeulen. A comparison of multivariate genome-wide association methods. PloS one, 9(4):e95923, 2014. doi: https://doi.org/10.1371/journal.pone.0095923.

Joanna Zhuang, Martin Widschwendter, and Andrew E Teschendorff. A comparison of feature selection and classification methods in dna methylation studies using the illumina infinium platform. BMC bioinformatics, 13:1–14, 2012. doi: https://doi.org/10.1186/1471-2105-13-59.

Cen Wu, Fei Zhou, Jie Ren, Xiaoxi Li, Yu Jiang, and Shuangge Ma. A selective review of multi-level omics data integration using variable selection. High-throughput, 8(1):4, 2019. doi: https://doi.org/10.3390/ht8010004.

Radford M Neal and Jianguo Zhang. High dimensional classification with bayesian neural networks and dirichlet diffusion trees. Feature extraction: Foundations and applications, pages 265–296, 2006. doi: https://doi.org/10.1007/978-3-540-35488-8_11.

Daniel Gianola, Hayrettin Okut, Kent A Weigel, and Guilherme JM Rosa. Predicting complex quantitative traits with bayesian neural networks: a case study with jersey cows and wheat. BMC genetics, 12:1–14, 2011. doi: https://doi.org/10.1186/1471-2156-12-87.

Bob Carpenter, Andrew Gelman, Matthew D Hoffman, Daniel Lee, Ben Goodrich, Michael Betancourt, Marcus Brubaker, Jiqiang Guo, Peter Li, and Allen Riddell. Stan: A probabilistic programming language. Journal

of statistical software, 76(1), 2017. doi: https://doi.org/10.18637/jss.v076.i01.

Shaun Purcell, Benjamin Neale, Kathe Todd-Brown, Lori Thomas, Manuel AR Ferreira, David Bender, Julian Maller, Pamela Sklar, Paul IW De Bakker, Mark J Daly, et al. Plink: a tool set for whole-genome association and population-based linkage analyses. The American journal of human genetics, 81(3):559–575, 2007. doi: 10.1086/519795.

Po-Ru Loh, George Tucker, Brendan K Bulik-Sullivan, Bjarni J Vilhjalmsson, Hilary K Finucane, Rany M Salem, Daniel I Chasman, Paul M Ridker, Benjamin M Neale, Bonnie Berger, et al. Efficient bayesian mixed-model analysis increases association power in large cohorts. Nature genetics, 47(3):284–290, 2015. doi: https://doi.org/10.1038/ng.3190.

Christoph Lippert, Jennifer Listgarten, Ying Liu, Carl M Kadie, Robert I Davidson, and David Heckerman. Fast linear mixed models for genome-wide association studies. Nature methods, 8(10):833–835, 2011. doi: https://doi.org/10.1038/nmeth.1681.

Jian Yang, S Hong Lee, Michael E Goddard, and Peter M Visscher. Gcta: a tool for genome-wide complex trait analysis. The American Journal of Human Genetics, 88(1):76–82, 2011. doi: https://doi.org/10.1016/j.ajhg.2010.11.011.

Peter J Bradbury, Zhiwu Zhang, Dallas E Kroon, Terry M Casstevens, Yogesh Ramdoss, and Edward S Buckler. Tassel: software for association mapping of complex traits in diverse samples. Bioinformatics, 23(19):2633–2635, 2007. doi: 10.1093/bioinformatics/btm308.

Alexander E Lipka, Feng Tian, Qishan Wang, Jason Peiffer, Meng Li, Peter J Bradbury, Michael A Gore, Edward S Buckler, and Zhiwu Zhang. Gapit: genome association and prediction integrated tool. Bioinformatics, 28(18): 2397–2399, 2012. doi: 10.1093/bioinformatics/bts444.

You Tang, Xiaolei Liu, Jiabo Wang, Meng Li, Qishan Wang, Feng Tian, Zhongbin Su, Yuchun Pan, Di Liu, Alexander E Lipka, et al. Gapit version 2: an enhanced integrated tool for genomic association and prediction. The plant genome, 9(2):plantgenome2015–11, 2016. doi: 10.3835/plantgenome2015.11.012.

Wanfang Fu, Cassia da Silva Linge, and Ksenija Gasic. Genome-wide association study of brown rot (monilinia spp.) tolerance in peach. Frontiers in Plant Science, 12:635914, 2021. doi: https://doi.org/10.3389/fpls.2021.635914.

Julie Sardos, Mathieu Rouard, Yann Hueber, Alberto Cenci, Katie E Hyma, Ines Van Den Houwe, Eva Hribova, Brigitte Courtois, and Nicolas Roux. A genome-wide association study on the seedless phenotype in banana (musa spp.) reveals the potential of a selected panel to detect candidate genes in a vegetatively propagated crop. PLoS One, 11(5):e0154448, 2016. doi: https://doi.org/10.1371/journal.pone.0154448.

Susanna Atwell, Yu S Huang, Bjarni J Vilhjálmsson, Glenda Willems, Matthew Horton, Yan Li, Dazhe Meng, Alexander Platt, Aaron M Tarone, Tina T Hu, et al. Genome-wide association study of 107 phenotypes in arabidopsis thaliana inbred lines. Nature, 465(7298):627–631, 2010. doi: 10.1038/nature08800.

Danelle K Seymour, Eunyoung Chae, Dominik G Grimm, Carmen Mar-

tin Pizarro, Anette Habring-Müller, François Vasseur, Barbara Rakitsch, Karsten M Borgwardt, Daniel Koenig, and Detlef Weigel. Genetic architecture of nonadditive inheritance in arabidopsis thaliana hybrids. Proceedings of the National Academy of Sciences, 113(46):E7317–E7326, 2016. doi: https://doi.org/10.1073/pnas.1615268113.

Richard Lewontin. Race - the power of an illusion., 2003. URL https://www.pbs.org/race/000_About/002_04-background-01-04.htm.

William S Bush and Jason H Moore. Chapter 11: Genome-wide association studies. PLoS computational biology, 8(12):e1002822, 2012. doi: https://doi.org/10.1371/journal.pcbi.1002822.

Francis Robert and Jerry Pelletier. Exploring the impact of single-nucleotide polymorphisms on translation. Frontiers in genetics, 9:507, 2018. doi: https://doi.org/10.3389/fgene.2018.00507.

Emil Uffelmann, Qin Qin Huang, Nchangwi Syntia Munung, Jantina De Vries, Yukinori Okada, Alicia R Martin, Hilary C Martin, Tuuli Lappalainen, and Danielle Posthuma. Genome-wide association studies. Nature Reviews Methods Primers, 1(1):1–21, 2021. doi: https://doi.org/10.1038/s43586-021-00056-9.

Daniel C Koboldt. Best practices for variant calling in clinical sequencing. Genome Medicine, 12(1):1–13, 2020. doi: https://doi.org/10.1186/s13073-020-00791-w.

Richard M Leggett and Dan MacLean. Reference-free snp detection: dealing with the data deluge. Bmc Genomics, 15(4):1–7, 2014. doi: https://doi.org/10.1186/1471-2164-15-S4-S10.

Rita M Cantor, Kenneth Lange, and Janet S Sinsheimer. Prioritizing gwas results: a review of statistical methods and recommendations for their application. The American Journal of Human Genetics, 86(1):6–22, 2010. doi: https://doi.org/10.1016/j.ajhg.2009.11.017.

M Kamran Ikram, Sim Xueling, Richard A Jensen, Mary Frances Cotch, Alex W Hewitt, M Arfan Ikram, Jie Jin Wang, Ronald Klein, Barbara EK Klein, Monique MB Breteler, et al. Four novel loci (19q13, 6q24, 12q24, and 5q14) influence the microcirculation in vivo. PLoS genetics, 6(10): e1001184, 2010. doi: https://doi.org/10.1371/annotation/841bfadf-85d1-4059-894f-2863d73fa963.

Donglei Hu and Elad Ziv. Confounding in genetic association studies and its solutions. Pharmacogenomics in Drug Discovery and Development, pages 31–39, 2008. doi: https://doi.org/10.1007/978-1-59745-205-2_3.

Zitong Li, Petri Kemppainen, Pasi Rastas, and Juha Merilä. Linkage disequilibrium clustering-based approach for association mapping with tightly linked genomewide data. Molecular Ecology Resources, 18(4):809–824, 2018a. doi: https://doi.org/10.1111/1755-0998.12893.

ME Goddard and BJ Hayes. Genomic selection. Journal of Animal Breeding and Genetics, 124(6):323–330, 2007. doi: https://doi.org/10.1111/j.1439-0388.2007.00702.x.

Jiabo Wang and Zhiwu Zhang. Gapit version 3: boosting power and accuracy for genomic association and prediction. Genomics, proteomics & bioinformatics, 19(4):629–640, 2021. doi: https://doi.org/10.1016/j.gpb.2021.08.005.

Zhongsheng Chen, Michael Boehnke, Xiaoquan Wen, and Bhramar Mukherjee. Revisiting the genome-wide significance threshold for common variant gwas. G3, 11(2):jkaa056, 2021. doi: 10.1093/g3journal/jkaa056.

Avjinder S Kaler and Larry C Purcell. Estimation of a significance threshold for genome-wide association studies. BMC genomics, 20:1–8, 2019. doi: https://doi.org/10.1186/s12864-019-5992-7.

Christopher C Chang. Data management and summary statistics with plink. Methods Mol Biol, 2090:49–65, 2020. doi: https://doi.org/10.1007/978-1-0716-0199-0_3.

A Richard, John W Gibbs, Paul Hardenbol Belmont, D Willis Thomas, Huanming Yang Fuli Yu, Wei Huang Lan-Yang Ch'ang, Yan Shen Bin Liu, et al. The international hapmap project. Nature, 426(6968):789–796, 2003. doi: http://dx.doi.org/10.1038/nature02168.

Matthew S Lyon, Shea J Andrews, Ben Elsworth, Tom R Gaunt, Gibran Hemani, and Edoardo Marcora. The variant call format provides efficient and robust storage of gwas summary statistics. Genome biology, 22(1): 1–10, 2021. doi: https://doi.org/10.1186/s13059-020-02248-0.

Layan Imad Nahlawi. Genetic feature selection using dimensionality reduction approaches: A comparative study. PhD thesis, Queen's University, 2010.

Jun Yan and Xiangfeng Wang. Machine learning bridges omics sciences and plant breeding. Trends in Plant Science, 2022. doi: https://doi.org/10.1016/j.tplants.2022.08.018.

Silke Szymczak, Joanna M Biernacka, Heather J Cordell, Oscar González-Recio, Inke R König, Heping Zhang, and Yan V Sun. Machine learning in

genome-wide association studies. Genetic epidemiology, 33(S1):S51–S57, 2009. doi: https://doi.org/10.1002/gepi.20473.

Bettina Mieth, Marius Kloft, Juan Antonio Rodríguez, Sören Sonnenburg, Robin Vobruba, Carlos Morcillo-Suárez, Xavier Farré, Urko M Marigorta, Ernst Fehr, Thorsten Dickhaus, et al. Combining multiple hypothesis testing with machine learning increases the statistical power of genome-wide association studies. Scientific reports, 6(1):36671, 2016. doi: https://doi.org/10.1038/srep36671.

Hamid Behravan, Jaana M Hartikainen, Maria Tengström, Katri Pylkäs, Robert Winqvist, Veli-Matti Kosma, and Arto Mannermaa. Machine learning identifies interacting genetic variants contributing to breast cancer risk: A case study in finnish cases and controls. Scientific reports, 8(1):13149, 2018. doi: https://doi.org/10.1038/s41598-018-31573-5.

Yingjie Guo, Chenxi Wu, Maozu Guo, Quan Zou, Xiaoyan Liu, and Alon Keinan. Combining sparse group lasso and linear mixed model improves power to detect genetic variants underlying quantitative traits. Frontiers in Genetics, 10:271, 2019. doi: https://doi.org/10.3389/fgene.2019.00271.

Jiahan Li, Zhong Wang, Runze Li, and Rongling Wu. Bayesian group lasso for nonparametric varying-coefficient models with application to functional genome-wide association studies. The annals of applied statistics, 9(2):640, 2015. doi: 10.1214/15-AOAS808.

Patrik Waldmann, Gábor Mészáros, Birgit Gredler, Christian Fuerst, and Johann Sölkner. Evaluation of the lasso and the elastic net in genome-wide association studies. Frontiers in genetics, 4:270, 2013. doi: https://doi.org/10.3389/fgene.2013.00270.

Sebastian Okser, Tapio Pahikkala, Antti Airola, Tapio Salakoski, Samuli Ripatti, and Tero Aittokallio. Regularized machine learning in the genetic prediction of complex traits. PLoS genetics, 10(11):e1004754, 2014. doi: https://doi.org/10.1371/journal.pgen.1004754.

Sudeep Srivastava and Liang Chen. Comparison between the ssvs and the lasso for genome-wide association studies. Commun. Inf. Syst., 10(2):39–52, 2010. URL http://dml.mathdoc.fr/item/1268143372.

Verena Zuber, A Pedro Duarte Silva, and Korbinian Strimmer. A novel algorithm for simultaneous snp selection in high-dimensional genome-wide association studies. BMC bioinformatics, 13:1–8, 2012. doi: https://doi.org/10.1186/1471-2105-13-284.

Gabriel E Hoffman, Benjamin A Logsdon, and Jason G Mezey. Puma: a unified framework for penalized multiple regression analysis of gwas data. PLoS computational biology, 9(6):e1003101, 2013. doi: https://doi.org/10.1371/journal.pcbi.1003101.

Adrien Badré, Li Zhang, Wellington Muchero, Justin C Reynolds, and Chongle Pan. Deep neural network improves the estimation of polygenic risk scores for breast cancer. Journal of Human Genetics, 66(4):359–369, 2021. doi: https://doi.org/10.1038/s10038-020-00832-7.

Yang Liu, Duolin Wang, Fei He, Juexin Wang, Trupti Joshi, and Dong Xu. Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. Frontiers in genetics, 10:1091, 2019. doi: https://doi.org/10.3389/fgene.2019.01091.

Alberto Romagnoni, Simon Jégou, Kristel Van Steen, Gilles Wainrib, and Jean-Pierre Hugot. Comparative performances of machine learning meth-

ods for classifying crohn disease patients using genome-wide genotyping data. Scientific reports, 9(1):10351, 2019. doi: https://doi.org/10.1038/s41598-019-46649-z.

Patrik Waldmann. Approximate bayesian neural networks in genomic prediction. Genetics Selection Evolution, 50:1–9, 2018. doi: https://doi.org/10.1186/s12711-018-0439-1.

Andrew L Beam, Alison Motsinger-Reif, and Jon Doyle. Bayesian neural networks for detecting epistasis in genetic association studies. BMC bioinformatics, 15:1–12, 2014. doi: https://doi.org/10.1186/s12859-014-0368-0.

R. Setiono and L.C.K. Hui. Use of a quasi-newton method in a feedforward neural network construction algorithm. IEEE Transactions on Neural Networks, 6(1):273–277, 1995. doi: 10.1109/72.363426.

Shuo Shi, NA Yuan, Ming Yang, Zhenglin Du, Jinyue Wang, Xin Sheng, Jiayan Wu, and Jingfa Xiao. Comprehensive assessment of genotype imputation performance. Human Heredity, 83(3):107–116, 2018. doi: https://doi.org/10.1159/000489758.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. Journal of statistical software, 33(1):1, 2010. doi: 10.18637/jss.

Yi Yang and Hui Zou. A fast unified algorithm for solving group-lasso penalize learning problems. Statistics and Computing, 25:1129–1141, 2015. doi: https://doi.org/10.1007/s11222-014-9498-5.

Noah Simon, Jerome Friedman, Trevor Hastie, Rob Tibshirani, and Maintainer Noah Simon. Package 'sgl'. CRAN Documentation, 2018.

John C Whittaker, Robin Thompson, and Mike C Denham. Marker-assisted selection using ridge regression. Genetics Research, 75(2):249–252, 2000. doi: 10.1017/s0016672399004462.

Ghadeer Jasim Mohammed Mahdi, Nadia Jasim Mohammed, and Zahraa Ibrahim Al-Sharea. Regression shrinkage and selection variables via an adaptive elastic net model. In Journal of Physics: Conference Series, volume 1879, page 032014. IOP Publishing, 2021. doi: 10.1088/1742-659 6/1879/3/032014.

Daniel Berrar. Cross-validation. In Shoba Ranganathan, Michael Gribskov, Kenta Nakai, and Christian Schönbach, editors, Encyclopedia of Bioinformatics and Computational Biology, pages 542–545. Academic Press, Oxford, 2019. ISBN 978-0-12-811432-2. doi: https://doi.org/10 .1016/B978-0-12-809633-8.20349-X.

Yoonsuh Jung and Jianhua Hu. Ak-fold averaging cross-validation procedure. Journal of nonparametric statistics, 27(2):167–179, 2015. doi: 10.1080/10 485252.2015.1010532.

Jason Brownlee. Tour of evaluation metrics for imbalanced classification, 2020.

Cyril Goutte and Eric Gaussier. A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings 27, pages 345–359. Springer, 2005. doi: https://doi.org/10.1007/978$-$3$-$5 40$-$31865$-$1_25.

Seong Ho Park, Jin Mo Goo, and Chan-Hee Jo. Receiver operating characteristic (roc) curve: practical review for radiologists. Korean journal of radiology, 5(1):11–18, 2004. doi: 10.3348/kjr.2004.5.1.11.

David Faraggi and Benjamin Reiser. Estimation of the area under the roc curve. Statistics in medicine, 21(20):3093–3106, 2002. doi: 10.1002/sim.1228.

Mohammad Hossin and Md Nasir Sulaiman. A review on evaluation metrics for data classification evaluations. International journal of data mining & knowledge management process, 5(2):1, 2015. doi: 10.5121/ijdkp.2015.5201.

Frank Nielsen and Frank Nielsen. Hierarchical clustering. Introduction to HPC with MPI for Data Science, pages 195–211, 2016. doi: https://doi.org/10.1007/978-3-319-21903-5_8.

Douglas M Hawkins. The problem of overfitting. Journal of chemical information and computer sciences, 44(1):1–12, 2004. doi: https://doi.org/10.1021/ci0342472.

R Team et al. Function chisq. test. R stats package, version, 1.3, 2020.

Shichao Jin, Yanjun Su, Shang Gao, Fangfang Wu, Tianyu Hu, Jin Liu, Wenkai Li, Dingchang Wang, Shaojiang Chen, Yuanxi Jiang, et al. Deep learning: individual maize segmentation from terrestrial lidar data using faster r-cnn and regional growth algorithms. Frontiers in plant science, 9:866, 2018. doi: https://doi.org/10.3389/fpls.2018.00866.

Nina Zhou and Lipo Wang. A modified t-test feature selection method and its application on the hapmap genotype data. Genomics, proteomics &

bioinformatics, 5(3-4):242–249, 2007. doi: https://doi.org/10.1016/S1672-0229(08)60011-X.

Xin Jin, Anbang Xu, Rongfang Bie, and Ping Guo. Machine learning techniques and chi-square feature selection for cancer classification using sage gene expression profiles. In Data Mining for Biomedical Applications: PAKDD 2006 Workshop, BioDM 2006, Singapore, April 9, 2006. Proceedings, pages 106–115. Springer, 2006. doi: https://doi.org/10.1007/11691730_11.

Shaikh Shakeela, N Sai Shankar, P Mohan Reddy, T Kavya Tulasi, and M Mahesh Koneru. Optimal ensemble learning based on distinctive feature selection by univariate anova-f statistics for ids. International Journal of Electronics and Telecommunications, pages 267–275, 2021. doi: 10.24425/ijet.2021.135975.

Anisha Arora, Arno Candel, Jessica Lanford, Erin LeDell, and Viraj Parmar. Deep learning with h2o. H2O. ai, Mountain View, 587, 2015.

Suneetha Uppu, Aneesh Krishna, and Raj P Gopalan. Towards deep learning in genome-wide association interaction studies. 2016. URL https://aisel.aisnet.org/pacis2016/20.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. nature, 521(7553):436–444, 2015. doi: https://doi.org/10.1038/nature14539.

José Miguel Hernández-Lobato and Ryan Adams. Probabilistic backpropagation for scalable learning of bayesian neural networks. In International conference on machine learning, pages 1861–1869. PMLR, 2015.

Peter D Hoff. A first course in Bayesian statistical methods, volume 580. Springer, 2009. doi: https://doi.org/10.1007/978-0-387-92407-6.

Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 4697–4708. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/322f62469c5e3c7dc3e58f5a4d1ea399-Paper.pdf.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. Advances in neural information processing systems, 30, 2017. doi: 10.48550/arXiv.1612.01474.

Radford M Neal et al. Mcmc using hamiltonian dynamics. Handbook of markov chain monte carlo, 2(11):2, 2011.

Christopher M Bishop and Nasser M Nasrabadi. Pattern recognition and machine learning, volume 4. Springer, 2006.

Matthew D Hoffman, Andrew Gelman, et al. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. J. Mach. Learn. Res., 15(1):1593–1623, 2014. doi: https://doi.org/10.48550/arXiv.1111.4246.

Michael Betancourt. A conceptual introduction to hamiltonian monte carlo. arXiv preprint arXiv:1701.02434, 2017. doi: https://doi.org/10.48550/arXiv.1701.02434.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. Journal of the American statistical Association, 112(518):859–877, 2017. doi: https://doi.org/10.1080/01621459.2017.1285773.

Charles E Brown. Coefficient of variation. In Applied multivariate statistics in geohydrology and related sciences, pages 155–157. Springer, 1998. doi: https://doi.org/10.1007/978-3-642-80328-4_13.

Arthur G Bedeian and Kevin W Mossholder. On the use of the coefficient of variation as a measure of diversity. Organizational Research Methods, 3(3):285–297, 2000.

Ömer Faruk Ertuğrul and Mehmet Emin Tağluk. A fast feature selection approach based on extreme learning machine and coefficient of variation. Turkish Journal of Electrical Engineering and Computer Sciences, 25(4): 3409–3420, 2017. doi: https://doi.org/10.3906/elk-1606-122.

Andrew Gelman, Daniel Lee, and Jiqiang Guo. Stan: A probabilistic programming language for bayesian inference and optimization. Journal of Educational and Behavioral Statistics, 40(5):530–543, 2015. doi: 10.3102/1076998615606113.

J Gabry. shinystan: Interactive visual and numerical diagnostics and posterior analysis for bayesian models (r package version 2.4. 0, 2017), 2020.

Chaewon Park, Jin-Woong Lee, Minseuk Kim, Byung Do Lee, Satendra Pal Singh, Woon Bae Park, and Kee-Sun Sohn. A data-driven approach to predicting band gap, excitation, and emission energies for eu 2+-activated phosphors. Inorganic Chemistry Frontiers, 8(21):4610–4624, 2021. doi: 10.1039/d1qi00766a.

John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. Journal of the Royal Statistical Society: Series A (General), 135 (3):370–384, 1972. doi: https://doi.org/10.2307/2344614.

Daozong Chen, Haidong Chen, Guoqiang Dai, Haimei Zhang, Yi Liu, Wenjie Shen, Bo Zhu, Cheng Cui, and Chen Tan. Genome-wide identification of r2r3-myb gene family and association with anthocyanin biosynthesis in brassica species. BMC genomics, 23(1):1–13, 2022. doi: https://doi.org/10.1186/s12864-022-08666-7.

Muhammad Waqas, Luqman Shahid, Komal Shoukat, Usman Aslam, Farrukh Azeem, and Rana Muhammad Atif. Role of dna-binding with one finger (dof) transcription factors for abiotic stress tolerance in plants. In Transcription factors for abiotic stress tolerance in plants, pages 1–14. Elsevier, 2020. doi: https://doi.org/10.1016/B978-0-12-819334-1.00001-0.

You-Huang Xiang, Jia-Jun Yu, Ben Liao, Jun-Xiang Shan, Wang-Wei Ye, Nai-Qian Dong, Tao Guo, Yi Kan, Hai Zhang, Yi-Bing Yang, et al. An α/β hydrolase family member negatively regulates salt tolerance but promotes flowering through three distinct functions in rice. Molecular Plant, 15(12): 1908–1930, 2022. doi: 10.1016/j.molp.2022.10.017.

Yong Luo and Hong Jiao. Using the stan program for bayesian item response theory. Educational and psychological measurement, 78(3):384–408, 2018. doi: 10.1177/0013164417693666.

Juan Luo, Xu Wang, Lei Feng, Yan Li, and Jun-Xian He. The mitogen-activated protein kinase kinase 9 (mkk9) modulates nitrogen acquisition and anthocyanin accumulation under a nitrogen-limiting condition in arabidopsis. Biochemical and biophysical research communications, 487(3): 539–544, 2017. doi: 10.1016/j.bbrc.2017.04.065.

Massimo Iorizzo, Pablo F Cavagnaro, Hamed Bostan, Yunyang Zhao, Jianhui Zhang, and Philipp W Simon. A cluster of myb transcription fac-

tors regulates anthocyanin biosynthesis in carrot (daucus carota l.) root and petiole. Frontiers in Plant Science, 9:1927, 2019. doi: https://doi.org/10.3389/fpls.2018.01927.

José M Franco-Zorrilla, Irene López-Vidriero, José L Carrasco, Marta Godoy, Pablo Vera, and Roberto Solano. Dna-binding specificities of plant transcription factors and their potential to define target genes. Proceedings of the National Academy of Sciences, 111(6):2367–2372, 2014. doi: https://doi.org/10.1073/pnas.1316278111.

Ildoo Hwang, Huei-Chi Chen, and Jen Sheen. Two-component signal transduction pathways in arabidopsis. Plant Physiology, 129(2):500–515, 2002. doi: 10.1104/pp.005504.

Yukio Nagano, Hirofumi Furuhashi, Takehito Inaba, and Yukiko Sasaki. A novel class of plant-specific zinc-dependent dna-binding protein that binds to a/t-rich dna sequences. Nucleic acids research, 29(20):4097–4105, 2001. doi: 10.1093/nar/29.20.4097.

Natsumi Maruta, Yuri Trusov, Daisuke Urano, David Chakravorty, Sarah M Assmann, Alan M Jones, and Jose R Botella. Gtp binding by arabidopsis extra-large g protein 2 is not essential for its functions. Plant physiology, 186(2):1240–1253, 2021. doi: 10.1093/plphys/kiab119.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, pages 1050–1059. PMLR, 2016. URL https://proceedings.mlr.press/v48/gal16.html.

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In International conference

on machine learning, pages 1613–1622. PMLR, 2015. URL https://proceedings.mlr.press/v37/blundell15.html.

Timo M Deist, Frank JWM Dankers, Gilmer Valdes, Robin Wijsman, I-Chow Hsu, Cary Oberije, Tim Lustberg, Johan van Soest, Frank Hoebers, Arthur Jochems, et al. Machine learning algorithms for outcome prediction in (chemo) radiotherapy: An empirical comparison of classifiers. Medical physics, 45(7):3449–3459, 2018. doi: https://doi.org/10.1002/mp.12967.

Zoubin Ghahramani. Probabilistic machine learning and artificial intelligence. Nature, 521(7553):452–459, 2015. doi: https://doi.org/10.1038/nature14541.

Wei Li, Xiatian Zhu, and Shaogang Gong. Harmonious attention network for person re-identification. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2285–2294, 2018b. doi: https://doi.org/10.48550/arXiv.1802.08122.

Junzhou Huang and Tong Zhang. The benefit of group sparsity. The Annals of Statistics, 38(4):1978 – 2004, 2010. doi: 10.1214/09-AOS778.

Estelle Kuhn, Catherine Matias, and Tabea Rebafka. Properties of the stochastic approximation em algorithm with mini-batch sampling. Statistics and Computing, 30(6):1725–1739, 2020. doi: https://doi.org/10.1007/s11222-020-09968-0.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In Proceedings of the 28th international conference on machine learning (ICML-11), pages 681–688, 2011.

Michail Tsagris and Ioannis Tsamardinos. Feature selection with the r package mxm. F1000Research, 7:1505, 2019.

Congwei Sun, Zhongdong Dong, Lei Zhao, Yan Ren, Ning Zhang, and Feng Chen. The wheat 660k snp array demonstrates great potential for marker-assisted selection in polyploid wheat. Plant Biotechnology Journal, 18(6): 1354–1360, 2020. doi: https://doi.org/10.1111/pbi.13361.

Li Xing, Mary L Lesperance, and Xuekui Zhang. Simultaneous prediction of multiple outcomes using revised stacking algorithms. Bioinformatics, 36 (1):65–72, 2020. doi: https://doi.org/10.1093/bioinformatics/btz531.

Appendices

Appendix A

PentaPen Results

Table A.1: Comparison of Ridge, LASSO, and Elastic Net without SNP Pooling for all the phenotypes. The performance metrics are recorded for both training and testing sets.

Phenotype	**Metrics**	**Data Split**	**All SNPs**		
			Ridge	**LASSO**	**Elastic Net**
Anthocyanin	**Precision**	Train	0.95	0.765	0.751
		Test	0.851	0.756	0.742
	Recall	Train	0.911	0.764	0.754
		Test	0.826	0.745	0.748
	F1-score	Train	0.975	0.766	0.759
		Test	0.907	0.755	0.742
	AUC	Train	1	0.942	0.977
		Test	0.895	0.897	0.959
Width	**R-squared**	Train	0.944	0.82	0.808
		Test	0.694	0.801	0.778
	RMSE	Train	0.255	0.241	0.253
		Test	0.453	0.263	0.263
DTF	**R-squared**	Train	0.909	0.758	0.755
		Test	0.679	0.604	0.625
	RMSE	Train	0.008	0.013	0.013
		Test	0.679	0.604	0.625
Germ	**Accuracy**	Train	1	0.942	0.977
		Test	0.789	0.879	0.903

Table A.2: Comparison of Group LASSO, SGL using all SNPs, and PentaPez for all the phenotypes. The performance metrics are recorded for both training and testing sets.

Phenotype	**Metrics**	**Data Split**	**All SNPs**		
			Group LASSO	**SGL**	**PentaPen**
Anthocyanin	**Precision**	Train	0.925	0.951	0.8494
		Test	0.77	0.769	0.75
	Recall	Train	0.884	0.933	0.806
		Test	0.793	0.788	0.7014
	F1-score	Train	0.931	0.998	0.827
		Test	0.773	0.772	0.7249
	AUC	Train	0.576	0.998	0.958
		Test	0.499	0.516	0.879
Width	**R-squared**	Train	0.455	0.948	0.8323
		Test	0.052	0.081	0.724
	RMSE	Train	1.586	0.275	1.224
		Test	4.665	5.002	1.45
DTF	**R-squared**	Train	0.738	0.735	0.866
		Test	0.235	0.299	0.753
	RMSE	Train	0.397	1.098	0.0157
		Test	3.996	4.334	0.0175
Germ	**Accuracy**	Train	0.528	0.947	0.772
		Test	0.359	0.679	0.741

Table A.3: Comparison of penalized methodologies using filtered SNPs as predictors for all the phenotypes. The performance metrics are recorded for both training and testing sets.

Phenotype	**Metrics**	**Data Split**	**Filtered SNPs**	
			Group LASSO	**SGL**
Anthocyanin	**Precision**	Train	0.796	0.977
		Test	0.769	0.789
	Recall	Train	0.801	0.981
		Test	0.771	0.792
	F1-score	Train	0.809	0.989
		Test	0.773	0.796
	AUC	Train	0.983	0.963
		Test	0.952	0.826
Width	**R-squared**	Train	0.759	0.941
		Test	0.701	0.786
	RMSE	Train	0.451	0.095
		Test	0.496	0.465
DTF	**R-squared**	Train	0.626	0.818
		Test	0.557	0.724
	RMSE	Train	0.017	0.009
		Test	0.061	0.184
Germ	**Accuracy**	Train	0.882	0.711
		Test	0.871	0.609

Appendix B

Deep Learning Results

Table B.1: Comparison of BayesDL with a deep learning model, CNN. The two-split test performance metrics are recorded for comparison.

Phenotype	**Metrics**	**Data Split**	**Neural Networks**	
			CNN	**BNN**
Anthocyanin	**Precision**	Split 1	0.9545	0.87
		Split 2	0.7334	0.87084
	Recall	Split 1	0.6363	0.8656
		Split 2	0.7097	0.8531
	F1-score	Split 1	0.7636	0.8677
		Split 2	0.7213	0.8607
	AUC	Split 1	0.9372	0.87492
		Split 2	0.8415	0.87492
Width	**R-squared**	Split 1	0.5503	0.8313
		Split 2	0.6305	0.8386
	RMSE	Split 1	1.9195	1.4254
		Split 2	2.0317	1.5794
DTF	**R-squared**	Split 1	0.2693	0.9308
		Split 2	0.9609	0.9422
	RMSE	Split 1	0.0029	0.0005
		Split 2	0.0073	0.0004
Germ	**Accuracy**	Split 1	0.8077	0.9241
		Split 2	0.7258	0.9334