UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

INTERPRETABLE DEEP NEURAL NETWORKS FOR MORE ACCURATE PREDICTIVE GENOMICS AND GENOME-WIDE ASSOCIATION STUDIES.

A DISSERTATION

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

DOCTOR OF PHILOSOPHY

By

ADRIEN BADRÉ

Norman, Oklahoma

2023

INTERPRETABLE DEEP NEURAL NETWORKS FOR MORE ACCURATE PREDICTIVE GENOMICS AND GENOME-WIDE ASSOCIATION STUDIES.

A DISSERTATION APPROVED FOR THE
SCHOOL OF COMPUTER SCIENCE

BY THE COMMITTEE CONSISTING OF

Dr. Chongle Pan (Chair)

Dr. Dean Hougen

Dr. Qi Cheng

Dr. Henry Neeman

Dr. Krithivasan Sankaranarayanan

Acknowledgments

I want to thank Dr. Chongle Pan for being my committee chair, advisor, co-author, mentor, and friend. His support, specifically during the pandemic, and tireless persistence helped me realize this work. None of my achievements would have been possible without him.

I would also like to thank my wife and the love of my life, Charissa, for her endless love and support through this journey. Without her, I would have collapsed many times. I also want to thank my parents, Jérôme and Sophie, my sisters, Aurore and Flore, and my family for their continuous support.

Next, I would like to thank Virginie Perez-Woods. She was an amazingly helpful academic coordinator. More than being a great friend, she was a french presence and launched my dissertation journey in the best possible way. I also want to address my gratitude to all the current staff at the CS department that helped me and guided me during this fantastic experience. Then, my gratitude goes to all my labmates, and I wish them the best for their Ph.D. journey.

Finally, I would like to thank the professors (Dr. Cheng, Dr. Hougen, Dr. Neeman and Dr. Sankaranarayanan) on my committee for their time and advice.

Table of Contents

List Of Tables

List Of Figures

Abstract

Genome-wide association studies (GWAS) and predictive genomics have become increasingly important in genetics research over the past decade. GWAS involves the analysis of the entire genome of a large group of individuals to identify genetic variants associated with a particular trait or disease. Predictive genomics combines information from multiple genetic variants to predict the polygenic risk score (PRS) of an individual for developing a disease.

Machine learning is a branch of artificial intelligence that has revolutionized various fields of study, including computer vision, natural language processing, and robotics. Machine learning focuses on developing algorithms and models that enable computers to learn from data and make predictions or decisions without being explicitly programmed. Deep learning is a subset of machine learning that uses deep neural networks to recognize patterns and relationships.

In this dissertation, we first compared various machine learning and statistical models for estimating breast cancer PRS. A deep neural network (DNN) was found to be the most effective, outperforming other techniques such as BLUP, BayesA, and LDpred. In the test cohort with 50% prevalence, the receiver operating characteristic curves area under the curves (ROC AUCs) were 67.4% for DNN, 64.2% for BLUP, 64.5% for BayesA, and 62.4% for LDpred. While BLUP, BayesA, and LDpred generated PRS that followed a normal distribution in the case population, the PRS generated by DNN followed a bimodal distribution. This allowed DNN to achieve a recall of 18.8% at 90% precision in the test cohort, which extrapolates to 65.4% recall at 20% precision in a general population. Interpretation of the DNN model identified significant variants that were previously overlooked by GWAS, highlighting their importance in predicting breast cancer risk.

We then developed a linearizing neural network architecture (LINA) that provided first-order and second-order interpretations on both the instance-wise and model-wise

levels, addressing the challenge of interpretability in neural networks. LINA outperformed other algorithms in providing accurate and versatile model interpretation, as demonstrated in synthetic datasets and real-world predictive genomics applications, by identifying salient features and feature interactions used for predictions.

Finally, it has been observed that many complex diseases are related to each other through common genetic factors, such as pleiotropy or shared etiology. We hypothesized that this genetic overlap can be used to improve the accuracy of polygenic risk scores (PRS) for multiple diseases simultaneously. To test this hypothesis, we propose an interpretable multi-task learning approach based on the LINA architecture. We found that the parallel estimation of PRS for 17 prevalent cancers using a pan-cancer MTL model was generally more accurate than independent estimations for individual cancers using comparable single-task learning models. Similar performance improvements were observed for 60 prevalent non-cancer diseases in a pan-disease MTL model. Interpretation of the MTL models revealed significant genetic correlations between important sets of single nucleotide polymorphisms, suggesting that there is a well-connected network of diseases with a shared genetic basis.

Chapter 1

Introduction

In this section, I discuss various concepts and methods that will be used later. First, we will talk about machine learning, then we will introduce the necessary genomics knowledge.

1.1 Machine Learning (ML)

Machine Learning is a subfield of Artificial Intelligence (AI) where machines learn a task without being explicitly programmed for it, according to Arthur Samuel (El Naqa and Murphy, 2015). In machine learning, models are created using algorithms that can learn from data and make predictions or decisions based on that data. The models are trained on historical data, and the goal is to make accurate predictions or decisions about new, unseen data.

1.1.1 Types of Machine Learning

In this subsection, we introduce the different types of machine learning.

1.1.1.1 Supervised Learning

Supervised learning takes a dataset with features and labels as input and learns their relationships, as shown in Figure 1.1.

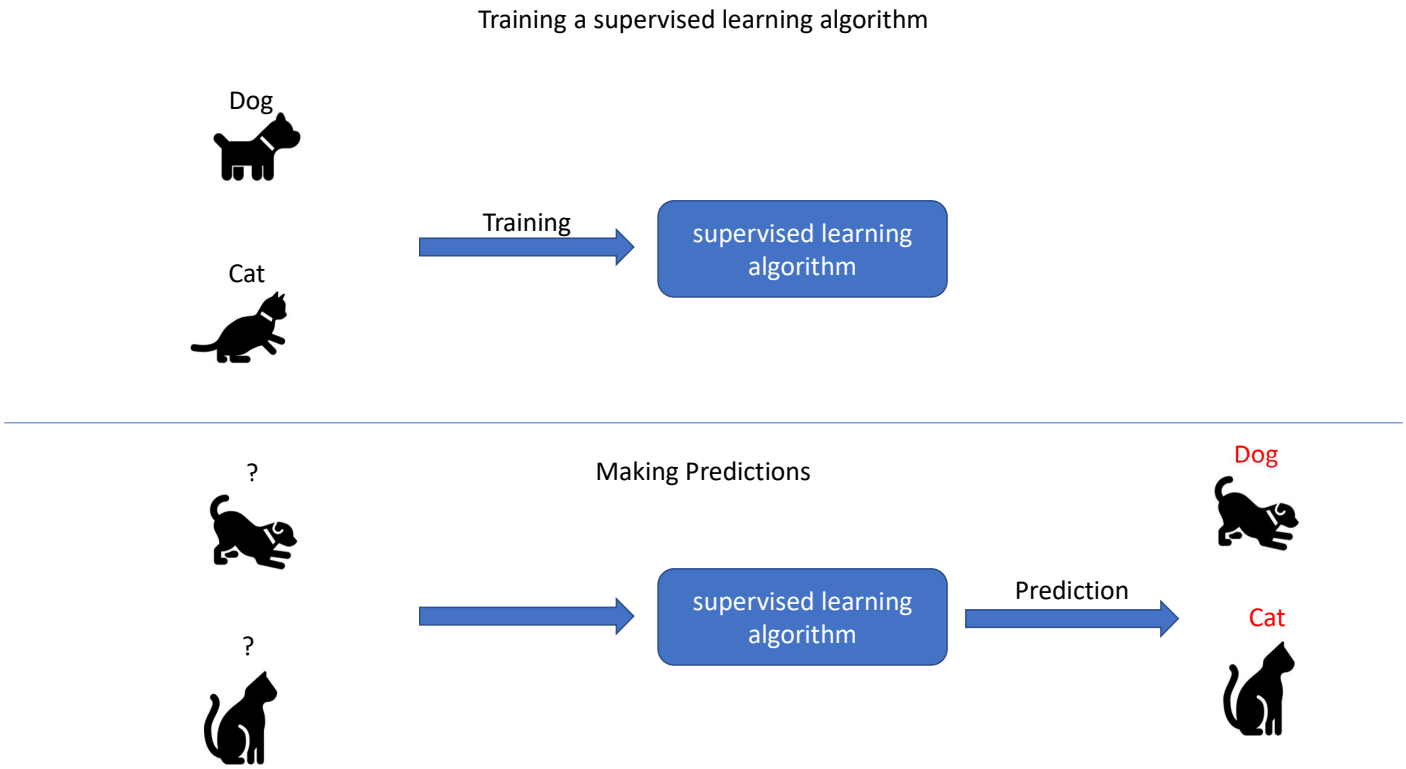


Figure 1.1: **Supervised learning algorithm trained to predict if an image is a cat or a dog.** The top of the picture shows what the algorithm is learning from. The bottom of the picture highlights the prediction task after the training phase: The algorithm takes unlabeled images, possibly containing a dog or a dog, and predicts a label for each of them.

More formally, let define $D = \{(x_1, y_1), ..., (x_i, y_i), .., (x_N, y_N)\}$ a dataset of size N. x_i is a feature vector of dimension d_x, and y_i its label. Let X be a matrix of feature vectors and Y a vector of labels, continuous or discrete. It is assumed that a relationship between X and Y exists. Let F be a model that has the capacity to learn the conditional probability $P(Y|X)$ and G a model that can learn the joint distribution $P(Y, X)$. In a supervised learning setup, the goal of F is to satisfy best a penalty function that models the bias/variance trade-off (Figure1.2), while G empirically seeks the function that best fits the training data.

Accurate learning of the relationships is measured by a loss function:

$$Loss_{total} = \sum_{i=0}^{N} L_i(y_i, M(x_i))$$

where M is the model and $M(x_i)$ is the score predicted by the model.

1.1.1.2 Unsupervised Learning

Unsupervised learning takes as input a dataset $D = \{(x_1), ..., (x_i), .., (x_N)\}$ without any label. Unlike supervised learning, where the algorithm is guided to represent a specific relationship between X and Y, the unsupervised learning algorithm needs to figure out by itself what relationship it needs to make inside the dataset and create its own labels, as demonstrated in Figure 1.3.

In this dissertation, we will focus on supervised learning.

1.1.2 Regression vs Classification

The supervised learning problems are divided into two categories: The regression tasks and the clarification tasks. Regression tasks can be defined as the process of finding the

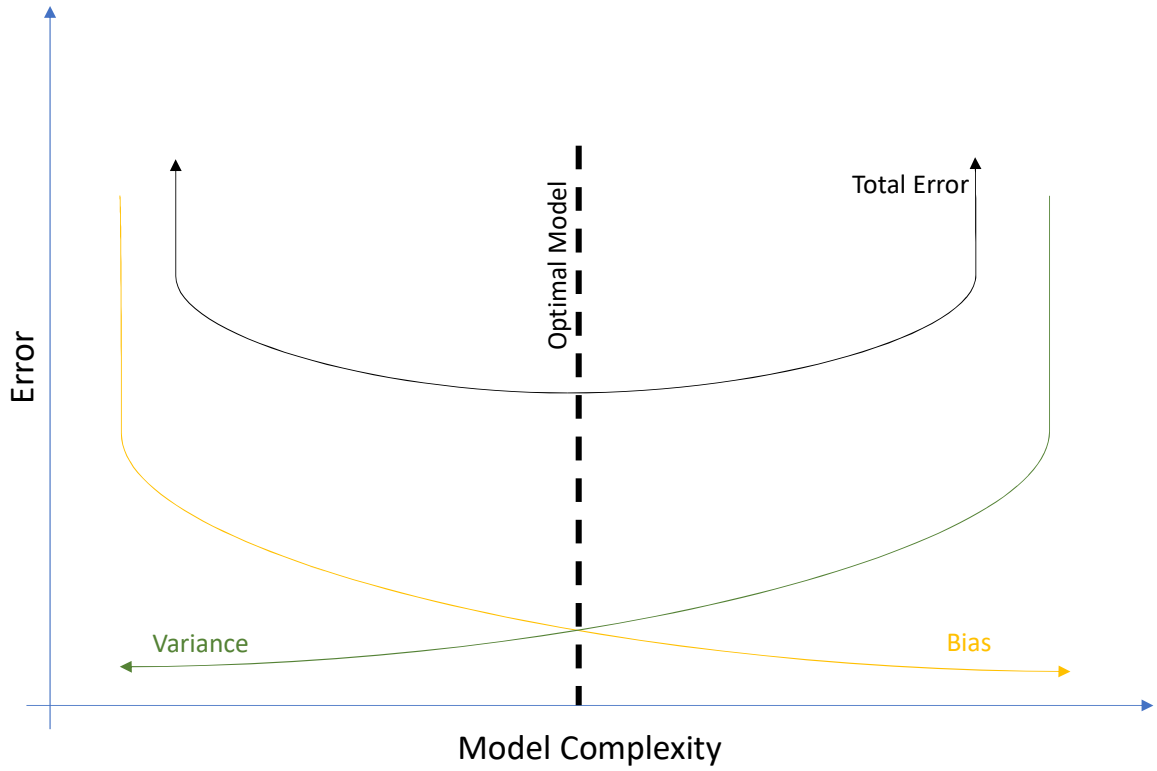


Figure 1.2: **The bias/variance trade-off.** The more the model becomes complex during its training phase, the more its bias decreases while the variance increases. The optimal point of learning is when the model variance and bias are the lowest because this is where the model total error reached the lowest value.

Figure 1.3: **An unsupervised learning algorithm trains to label the data points by itself.** The left side shows a dataset of unlabeled data points. The right side demonstrates that the model decided to make 3 groups of data points to describe the data it was given (the orange, green and gold groups).

relationship between X and Y where Y is continuous. A classification task, however, is a process of finding the relationship between X and Y where Y is discrete. Figure 1.4 illustrates the two types of distinct tasks.

1.1.3 Interpretable Machine Learning

The concept of interpretability in mathematics is not well defined. According to (Miller, 2019), interpretability can be defined as the extent to which a human can understand the reasoning behind a decision made by a model. Another definition is the extent to which a human can accurately predict the model's output. The more interpretable a machine learning model is, the easier it becomes for humans to comprehend its predictions or decisions. A model can be considered more interpretable if its decisions are easier for humans to understand. The terms interpretable and explainable are used interchangeably in this context. In this dissertation, interpretable machine learning refers to gaining meaningful insights from a machine learning model, whether the relationships are present in the data or learned by the model.

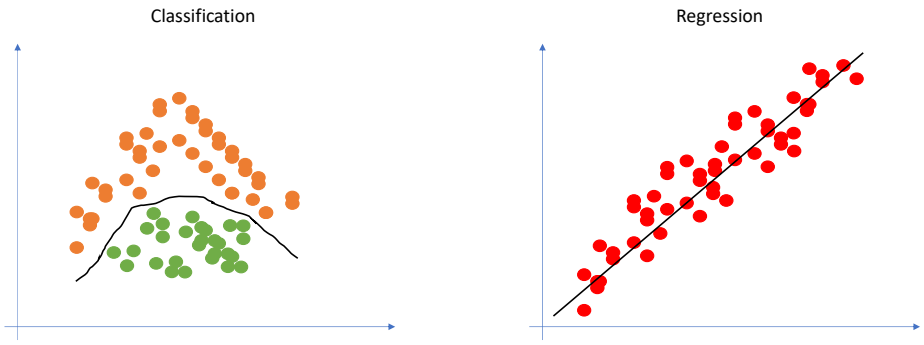


Figure 1.4: **Classification task versus a regression task** The left side shows that the model tries to draw a line to separate the green points from the orange points to classify them as accurately as possible. The right side demonstrates that the model is trying to draw a line to estimate the red points as closely as possible to realize a regression.

1.2 Supervised Machine Learning Algorithms

In this section, we introduce the necessary knowledge to understand the supervised deep neural network architectures used in this dissertation. First, I will describe the fundamental of linear regression and logistic regression. Then I will introduce neural networks and the notion of supervised learning.

1.2.1 Linear Regression

For a regression task,let's define a dataset $D = \{(x_1, y_1), ..., (x_i, y_i), .., (x_N, y_N)\}$, as previously defined. The labels Y are continuous, and the dataset carries the following assumptions:

- Features in X are independent from each other
- There is a linear relationship between X and Y
- Homoscedasticity: The variance of residual is the same for any value of X.
- For any fixed value of X, the mean of Y is normally distributed.

The relationship between the d features of x_i and the response y_i are modeled as follow:

$$y_i = \beta_d x_i^d + \beta_{d-1} x_i^{d-1} + ... + \beta_1 x_i^1 + \beta_0 + \epsilon_i \tag{1.1}$$

where the β_is represents the weights. There are $d + 1$ weights: one for each dimension, and β_0 is the bias. ϵ_i is the error term.

The Mean Square Error (MSE) is a common loss function that models the bias/variance trade-off. It is derived as:

$$MSE = \frac{1}{N} \sum_{i=0}^{N} (y_i - M(x_i))^2 \tag{1.2}$$

where M is our linear regression model.

The objective function is:

$$argmin_M(MSE) = \frac{1}{N}\sum_{i=0}^{N}(y_i - M^*(x_i))^2 \tag{1.3}$$

Where M^* is the optimal model M that minimizes the best MSE.

1.2.2 Logistic Regression

For a classification task,let's define a dataset $D = \{(x_1, y_1), ..., (x_i, y_i), .., (x_N, y_N)\}$, as previously defined. The labels Y are binary, and the dataset carries the following assumptions:

- Features in X are independent from each other
- There is a linear relationship between X and the response logit.
- For any fixed value of X, the Y is normally distributed.

The logistic regression model can be derived as follows:

$$logit = \log\frac{p}{1-p} = \beta_d x_i^d + \beta_{d-1} x_i^{d-1} + ... + \beta_1 x_i^1 + \beta_0 + \epsilon_i \tag{1.4}$$

$$p = \frac{1}{1 + e^{-logit}}$$

where p is the probability of a positive outcome.The function $f(x) = \frac{1}{1+e^{-x}}$ is called the sigmoid function and was introduced by Pierre François Verhulst in the 19th century.

The log-loss is a common loss function that models how close the prediction probability is to the actual binary values. It is derived as:

$$log_{loss} = \frac{1}{N}\sum_{i=0}^{N} y_i * \log M(x_i) + (1 - y_i) \log (1 - M(x_i)) \tag{1.5}$$

where M is our logistic regression model.

The objective function is:

$$argmin_M(log_{loss}) = \frac{1}{N}\sum_{i=0}^{N} y_i \log(M^*(x_i)) + (1 - y_i) \log(1 - M^*(x_i)) \tag{1.6}$$

Where M^* is the optimal model M that minimizes the best the log-loss.

1.2.3 Deep Learning

In this subsection, we will extend the supervised classification task to Artificial Neural Networks and introduce the concept of Multi-Task Learning (MTL).

1.2.3.1 Artificial Neural Network

History: Artificial neural networks (ANN) are complex statistical models inspired by biological neural networks. They are composed of neurons designed on the model of biological neurons (Figure 1.5). The artificial neurons take a signal as input from the preceding neurons' weighted outputs and get activated; based on the sum of these outputs. The function that decides whether the neuron gets activated is called the activation function. Like a biological neuron, if the strength of the input signal is strong enough, the neuron is activated and delivers a signal to the rest of the network it is connected to. If the connection between two neurons is deemed of importance,

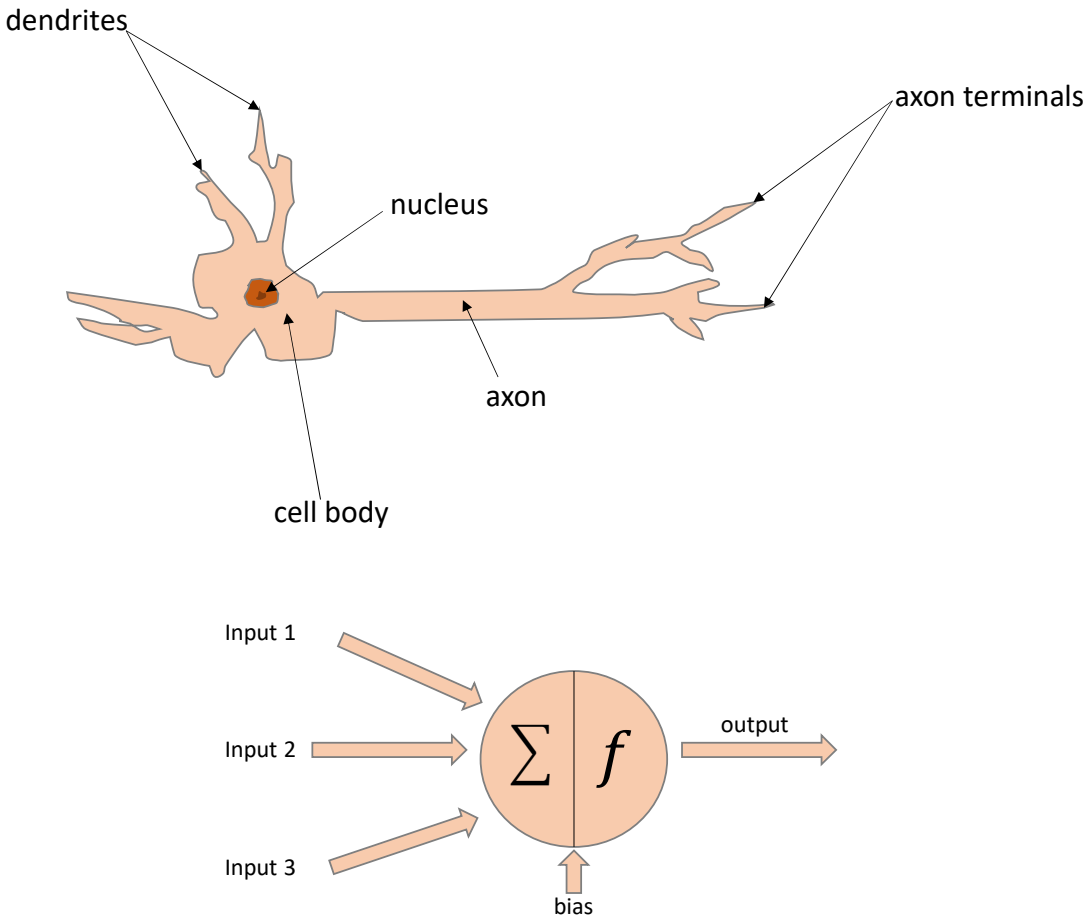


Figure 1.5: **A biological neuron (up) vs an artificial neuron(down).** The structure of the artificial neuron is very similar to the biological neuron. They both share a structure to receive an input signal and, based on its strength, decide to produce a corresponding output signal.

then, as with biological networks, the connection is reinforced. This gave birth to the perceptron by Franck Rosenblatt in 1958 (Rosenblatt, 1958). Artificial neural networks are constructed using layers of neurons. One layer contains several neurons, takes input from a preceding layer, and connects its outputs to another layer. When the neurons of each network layer are connected to all the neurons of the preceding layer and the next layer, it is called a fully connected layer (Figure 1.6). A network constructed with such layers is called a Feed Forward Neural Network (FFNN).

Activation functions: In a neuron, the strength of the input signal depends on the weights attributed to each connection. A neuron has one weight $w_{n,i,l}$ for each input connection. The output of a neuron n on layer l, given d neurons on the previous layer is given by:

$$output_{n,l} = f(\sum_{i=1}^{d} w_{n,i,l-1} output_{i,l-1}) \tag{1.7}$$

where f is the activation function of the neuron n. Several activation functions were developed. I am going to introduce the one I use in this dissertation. The first activation function presented here is sigmoid (Equation 1.4). In this dissertation, this function is used to activate the output neuron and map a real number to a probability (See Figure 1.7). Formally, for $x \in \mathbb{R}$, $sigmoid(x) \in [0, 1]$. Another activation function used in this dissertation is ReLU (Fukushima, 1975). The ReLU function is defined as follows:

$$ReLU(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{1.8}$$

This function is mainly used for neurons on hidden layers (layers that are between the input layer and the output layer) and introduces non-linearity to the network.

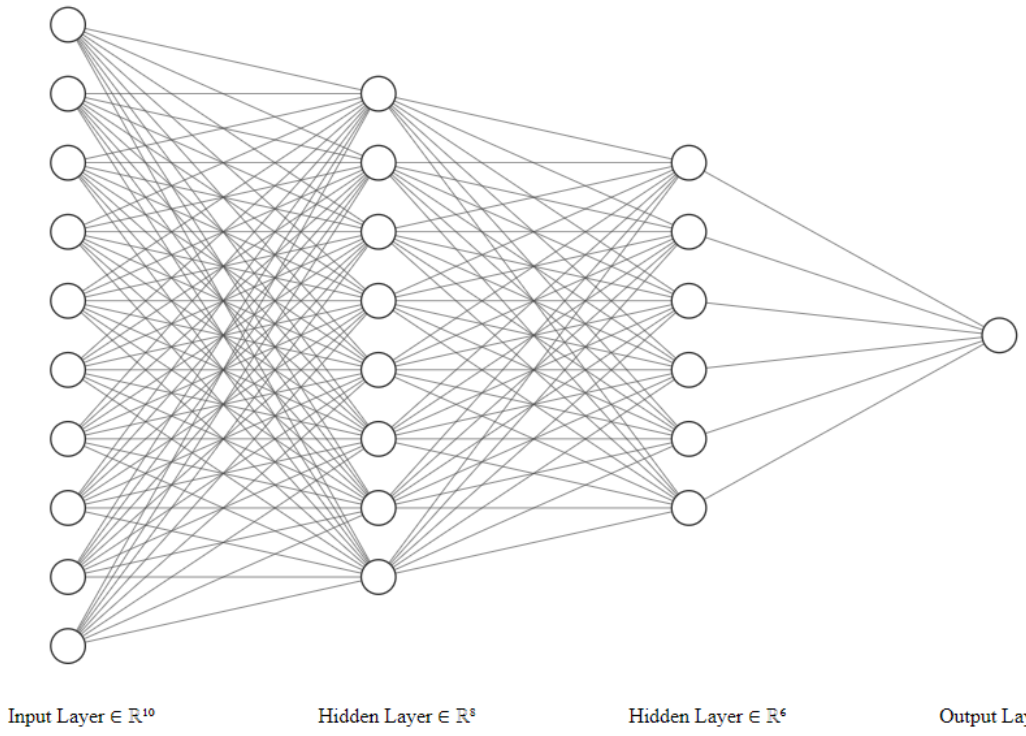


Figure 1.6: **A feed-forward neural network.** Each neuron is connected to all the neurons of the previous layer and the neurons of the next layer. Image is drawn using `https://alexlenail.me/NN-SVG/`.

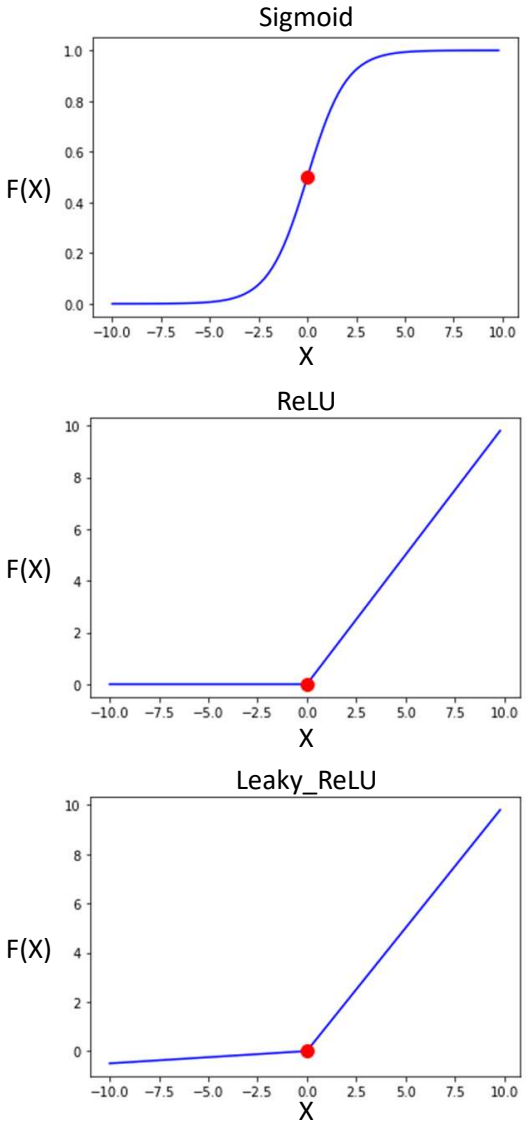


Figure 1.7: **The sigmoid curve, the ReLU curve, and the Leaky ReLU curve.**

However, the dying neuron is a drawback inherent to ReLU (Lu et al., 2019). Leaky ReLU was proposed to solve this issue (Maas et al., 2013). It allows the negative signals to be output with a small coefficient α. Formally it is defined as:

$$Leaky\_ReLU(x) = \begin{cases} x, & \text{if } x \geq 0 \\ -\alpha x, & \text{otherwise} \end{cases} \tag{1.9}$$

Training with Backpropagation: Training a FFNN for a supervised learning task requires the same setup as the linear regression for regression tasks or the logistic regression for classification tasks.

Let's define again a dataset$D = \{(x_1, y_1), ..., (x_i, y_i), .., (x_N, y_N)\}$ a dataset of size N. x_i is a feature vector of dimension d_x, and y_i its label. Let X be a matrix of feature vectors and Y a vector of continuous labels. The neural network F takes X as input and must learn the relationship between X and Y. The training goal is to minimize the loss function $\mathcal{L}$. Hence, the network must go through learning steps. At each learning step, a forward pass is realized. The network makes predictions using the input data. Then, the loss is measured to quantify the error. Finally, this error is backpropagated to the network that adapts its weights to minimize this error (Rumelhart et al., 1986). The algorithm is detailed in Algorithm 1.2.1 . In this dissertation, the Adam optimizer (Kingma and Ba, 2014a) version of the stochastic gradient update is used that introduce an adaptive estimate of lower-order moments and result in a faster and better gradient update algorithm.

Example of deep neural network architectures: Deep Neural Networks (DNN) are an ANN type that has more than 2 hidden layers. DNNs have been widely studied in the past years. They have met tremendous success in a vast number of different tasks,

Algorithm 1.2.1 Error backpropagation to the i_{th} weight on neuron n, on layer l, $w_{n,i,l}$ in a feed-forward neural network, at learning step t.

(1) $\forall(x_i, y_i) \in D$, propagate the input x_i through the network to compute the outputs Y^p, the vector containing all the y_i^p.

(2) Compute the loss $\mathcal{L}(Y^p, Y)$, with Y the vector containing all the y_i

(3) For each weight $w_{n,i,l}$ compute

$$w_{n,i,l}^{t+1} = w_{n,i,l}^{t} - \alpha \frac{\partial \mathcal{L}(Y^p, Y)}{\partial w_{n,i,l}} \tag{1.10}$$

with α referring to the learning rate, and t the learning step.

including audio and speech processing, visual data processing, and natural language processing (NLP) (Adeel et al., 2020; Tian et al., 2020; Young et al., 2018; Koppe et al., 2021) . Among them, Convolutional Neural Networks (CNN) (LeCun et al., 1998) have been part of these tremendous successes. CNNs perform pointwise multiplication of the input features with a moving filter across the feature space. Let's call each unit computation position the offset τ. The convolution operation for 1d datasets, as used in this dissertation, is computed as follows:

$$Conv(X, W, \tau) = \sum_{i=-\infty}^{\infty} x_i w_{\tau - i} \tag{1.11}$$

where X has d features, W is the kernel function (or weights of the neural network), x_i is a unique feature vector, and w_i is a weight in the filter. Figure 1.8 shows a 1D CNN. Multiple filters can be applied at each feature to decompose the signal into higher-order features. After the convolution operator is applied, the extracted feature space can be condensed using the pooling operator. Several blocks of convolution and pooling can be added together to condense the extracted feature. Then, those features can be flattened and passed to a FFNN to make the final prediction.

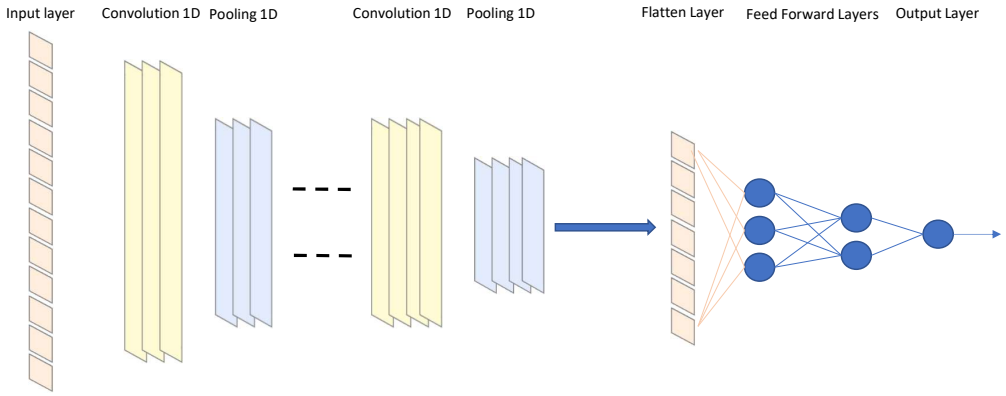


Figure 1.8: **Example of a CNN for object classification with 1D signal.** The signal is processed through several blocks of convolution and pooling, then the features are flattened to be processed by a FFNN for the final classification prediction.

1.2.3.2 Multi-task learning

Multitask Learning (Caruana, 1998) is a transfer learning method that enhances one task's performance by utilizing the information gained from related tasks. It accomplishes this by simultaneously training on multiple tasks while sharing a common representation, allowing what is learned from one task to improve the learning of other tasks. Multi-task learning was used with success in several different problems (Zhang and Yang, 2018), such as natural language processing (Collobert and Weston, 2008), speech recognition (Deng et al., 2013) or computer vision (Girshick, 2015). Formally, let's define a dataset $D = \{(x_1, y_1), ..., (x_i, y_i), .., (x_N, y_N)\}$ a dataset of size N. x_i is a feature vector of dimension d_x, and y_i its vector labels of dimension d_y. Each dimension of y_i refers to the label of x_i for task T_i. Let X be a matrix of feature vectors and Y a matrix of labels, Y_{T_i} being the label vector for task T_i. The network F takes X as input and must learn the relationship between X and Y. The network goal is to minimize the loss function $\mathcal{L}$ that is defined as follows:

$$Loss_{total} = \sum_{i=0}^{d_y} w_{T_i} L_i(M(X)_{T_i}, Y_{T_i}) \tag{1.12}$$

with w_i being the contribution weight of task T_i to the global loss, and $M(X)_{T_i}$ the model output for task T_i.

1.3 Genomics

Genomics encompasses the study of all genes in the genome, the interactions among genes, the genetic mutations, and the effects of genetic variants on human traits, known as phenotype. Mutations in an individual's genome can lead to dramatic changes in their phenotypes. Single Nucleotide Polymorphisms (SNP) represent substitutions of a

single nucleotide in the human genome. Figure 1.9 represents a simplified chromosome structure where those SNPs are quantified.

Different methods can be used to identify SNPs, such as dynamic allele-specific hybridization (Jobs, 2001), molecular beacons (Abravaya et al., 2003), and SNP microarrays (Steemers and Gunderson1, 2005; Thissen et al., 2019). SNP microarrays were notably used to sequence SNPs for the Oncorarray consortium (Amos et al., 2017), UKBiobank genomics data (Bycroft et al., 2018), and the 1000 Genome project data (Consortium et al., 2015).Sequenced personal genomes are compared with a reference genome that contains the most common variants at each locus within the population. Therefore, a dataset is created where each position can take 3 different values, as illustrated in Figure 1.10. For individual i at position j, we have:

$$SNP_{i,j} = \begin{cases} 0, & \text{if both maternal and paternal base pair at position } j \text{ mutated} \\ 1, & \text{if only one mutated} \\ 2, & \text{if no mutation} \end{cases} \tag{1.13}$$

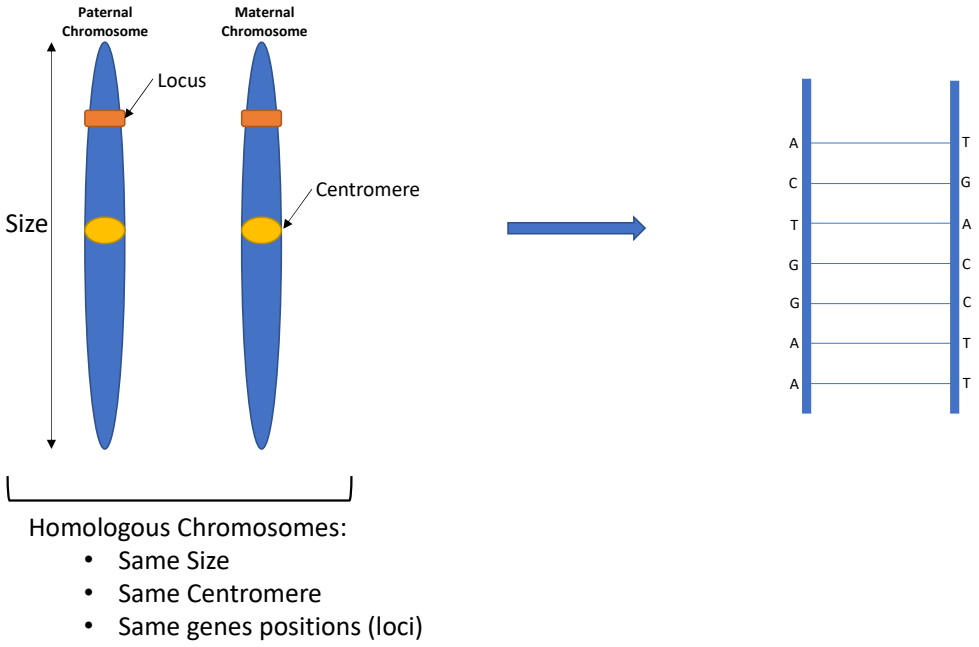


Figure 1.9: **Representation of a human chromosome** Each individual possesses a paternal chromosome and a maternal chromosome, linked through the centromere. The structure of those chromosomes is a double helix. There are 4 possible nucleotides: Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). A and T and C and G are exclusively bounded together on the double helix.

Genome

	SNP1	SNP2	SNP3	SNP4	SNP5	SNP6	SNP7	SNP8	SNP9...
Reference	A	T	C	G	A	A	A	C	T
Paternal	A	T	C	A	C	A	A	C	T
Maternal	A	T	C	A	A	A	A	C	T

Data

	SNP1	SNP2	SNP3	SNP4	SNP5	SNP6	SNP7	SNP8	SNP9...
Individual	2	2	2	0	1	2	2	2	2

Figure 1.10: **Mapping between the observed SNP and its data representation.** The red coloring shows a locus where 1 mutation happened on the maternal side, while the gold coloring shows a locus where 2 mutations happened. There is no distinction between the unique mutation on the maternal and paternal chromosomes.

1.4 Genome-Wide Association Studies

Genome-Wide Association Studies (GWAS) map SNP arrays to a trait to unveil the associations of variants with this particular trait. GWAS are generally conducted on a population sourced from a biobank, such as the UKBiobank (Bycroft et al., 2018), or study cohorts for specific diseases (Mailman et al., 2007). Human subjects are recruited on a volunteer-based system where they are asked to transmit their medical history. In some cases, such as UKBiobank, they are followed up throughout their life for potential additional traits developing with age.

Several data processing methods can be used to process the data, such as the minor allele frequency criteria, the Hardy–Weinberg equilibrium, or linkage disequilibrium. Hardly-Weinberg equilibrium relates to the principle that genetic variations stay the same from one generation to another. In this case, chi-square tests 1.14 are applied between an expected genetic population versus the current actual population. The test is formulated as follows:

$$\chi^2 = \sum_i \frac{(O_i - E_i)^2}{E_i} \tag{1.14}$$

E_i being the expected value at sample i and O the observed value at sample i.

If the test indicates a statistical difference between the expected population and the observed population, then the observed SNP genetic structure is in disequilibrium. Disequilibrium can indicate a significant amount of mutation rate, or non-random mating for example. Linkage disequilibrium refers to the non-random correlation between SNPs. Minor allele frequency refers to the frequency of the recessive allele for on position among the population. PLINK is the most popular software to assess those principles and manage this type of genomic data (Purcell et al., 2007).

Ancestry is also an important feature to consider because it may introduce bias in detecting variants that can lead to false positive variants over a population (Marchini

et al., 2004; Novembre et al., 2008; Lawson et al., 2020). Simple linear models may struggle to separate sub-populations effectively.

Statistical models have been widely used to model the relationships between SNPs and traits. Association analysis can be conducted using logistic regression between each SNP individually and the trait of interest. For each association, the p-value is calculated, and adjusted for multiple comparison using the Bonferroni level of significance (Bonferroni, 1935). The adjusted p-values are then commonly used to filter out significant SNPs from the non-important ones. Polygenic risk scores (PRS) can be derived using the additive effect of the SNPs. Linear regression models, such as Best Linear Unbiased Prediction (BLUP) (Henderson, 1975), consider the additive effects of SNPs to determine the relative importance of those SNPs. The genetic effect of SNPs is also associated with non-fixed effects, such as weight and environmental or behavioral factors. The model is structured as follows:

$$Y = W\alpha + X_s\beta_s + g + e \tag{1.15}$$

$$g \sim N(0, \sigma_a^2)$$

$$e \sim N(0, \sigma_e^2)$$

where σ_a^2 represents the genetic variation, σ_e^2 the residual variance for non-fixed effects, W is the covariates matrix for non-fixed effect, α its weight vector, X_s contains the SNPs matrix, and β_s the SNPs weights (Uffelmann et al., 2021).

1.5 Conclusion

In this section, I introduced several concepts used in this dissertation. The principle of machine learning, supervised learning, interpretable machine learning, multi-task

learning, neural networks as well as genomics, SNPs, GWAS, and PRS scores were covered. In this dissertation, we leverage the predicting power of neural networks, interpretable machine learning, and multi-task learning to redefine GWAS and enhance PRS score computation.

Chapter 2

Deep neural network improves the estimation of polygenic risk scores for breast cancer

In this chapter, I demonstrate the existence of non linear relationships between the genotype and the phenotype for breast cancer. This non-linearity is leveraged by a Deep Neural Network to achieve improved performance, compared to the baseline, for Breast Cancer PRS prediction accuracy.

2.1 Introduction

Breast cancer is the second deadliest cancer for U.S. women. Approximately one in eight women in the U.S. will develop invasive breast cancer over the course of their lifetime (NIH, 2012). Early detection of breast cancer is an effective strategy to reduce the death rate. If breast cancer is detected in the localized stage, the 5-year survival rate is 99% (NIH, 2012). However, only 62% of the breast cancer cases are detected in the localized stage (NIH, 2012). In 30% of the cases, breast cancer is detected after it spreads to the regional lymph nodes, reducing the 5-year survival rate to 85%. Furthermore, in 6% of cases, the cancer is diagnosed after it has spread to a distant part of the body beyond the lymph nodes and the 5-year survival rate is reduced to 27%. To detect breast cancer early, the US Preventive Services Task Force (USPSTF) recommends biennial screening mammography for women over 50 years old. For women under 50 years old, the decision for screening must be individualized to balance the benefit of potential early detection against the risk of a false positive diagnosis. False-positive mammography results, which typically lead to unnecessary follow-up diagnostic testing, become increasingly common for women 40 to 49 years old (Nelson et al., 2009). Nevertheless, for women with a high risk for breast cancer (i.e. a lifetime risk of breast cancer higher than 20%), the American Cancer Society advises a yearly breast MRI and mammogram starting at 30 years of age (Oeffinger et al., 2015).

Polygenic risk scores (PRS) assess the genetic risks of complex diseases based on the aggregate statistical correlation of a disease outcome with many genetic variations over the whole genome. Single-nucleotide polymorphisms (SNPs) are the most commonly used genetic variations. While genome-wide association studies (GWAS) report only SNPs with statistically significant associations to phenotypes (Dudbridge, 2013), PRS

can be estimated using a greater number of SNPs with higher adjusted p-value thresholds to improve prediction accuracy. Previous research has developed a variety of PRS estimation models based on Best Linear Unbiased Prediction (BLUP), including gBLUP (Clark et al., 2013), rr-BLUP (Whittaker et al., 2000a), (Meuwissen et al., 2001), and other derivatives (Maier et al., 2015; Speed and Balding, 2014). These linear mixed models consider genetic variations as fixed effects and use random effects to account for environmental factors and individual variability. Furthermore, linkage disequilibrium was utilized as a basis for the LDpred (Vilhjálmsson et al., 2015), (Khera et al., 2018), and PRS-CS (Ge et al., 2019) algorithms.

PRS estimation can also be defined as a supervised classification problem. The input features are genetic variations, and the output response is the disease outcome. Thus, machine learning techniques can be used to estimate PRS based on the classification scores achieved (Ho et al., 2019). A large-scale GWAS dataset may provide tens of thousands of individuals as training examples for model development and benchmarking. Wei et al (2019)(Wei et al., 2009) compared support vector machine and logistic regression to estimate PRS of Type-1 diabetes. The best Area Under the receiver operating characteristic Curve (AUC) was 84% in this study. More recently, neural networks have been used to estimate human height from the GWAS data, and the best R^2 scores were in the range of 0.4 to 0.5 (Bellot et al., 2018). Amyotrophic lateral sclerosis was also investigated using Convolutional Neural Networks (CNN) with 4511 cases and 6127 controls (Yin et al., 2019) and the highest accuracy was 76.9%.

Significant progress has been made in estimating PRS for breast cancer from a variety of populations. In a recent study (Mavaddat et al., 2019), multiple large European women cohorts were combined to compare a series of PRS models. The most predictive model in this study used lasso regression with 3,820 SNPs and obtained an AUC of 65%. A PRS algorithm based on the sum of log odds ratios of important SNPs

for breast cancer was used in the Singapore Chinese Health Study (Chan et al., 2018) with 46 SNPs and 56.6% AUC, the Shanghai Genome-Wide Association Studies (Wen et al., 2016) with 44 SNPs and 60.6% AUC, and a Taiwanese cohort (Hsieh et al., 2017) with 6 SNPs and 59.8% AUC. A pruning and thresholding method using 5,218 SNPs reached an AUC of 69% for the UK Biobank dataset (Khera et al., 2018).

In this study, deep neural network (DNN) was tested for breast cancer PRS estimation using a large cohort containing 26053 cases and 23058 controls. The performance of DNN was shown to be significantly higher than alternative machine learning algorithms and other statistical methods in this large cohort. Furthermore, DeepLift (Shrikumar et al., 2017) and LIME (Ribeiro et al., 2016) were used to identify salient SNPs used by DNN for prediction.

2.2 Methods

In this section , we detail the development of the workflow and our DNN model to improve PRS prediction, leveraging non linearity. We also detail the benchmark protocol with SOTA algortihms.

2.2.1 Breast cancer GWAS data

This study used a breast cancer GWAS dataset generated by the Discovery, Biology, and Risk of Inherited Variants in Breast Cancer (DRIVE) project (Amos et al., 2017) and was obtained from the NIH dbGaP database under the accession number of phs001265.v1.p1. The DRIVE dataset was stored, processed, and used on the Schooner supercomputer at the University of Oklahoma in an isolated partition with restricted access. The partition consisted of 5 computational nodes, each with 40 CPU cores

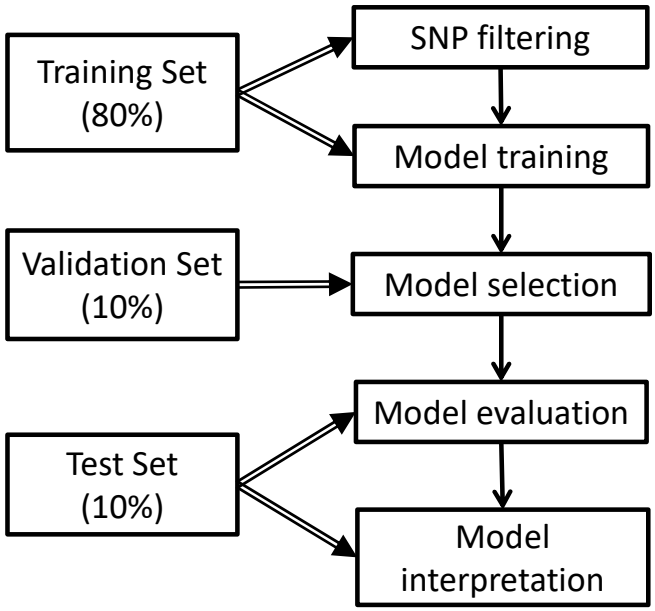


Figure 2.1: **Computational workflow of predictive genomics.** The DRIVE dataset was randomly split into the training set, the validation set, and the test set. Only the training set was used for association analysis, which generated the p-values for the selection of SNPs as input features. The training data was then used to train machine learning models and statistical models. The validation set was used to select the best hyperparameters for each model based on the validation AUC score. Finally, the test set was used for performance benchmarking and model interpretation.

(Intel Xeon Cascade Lake) and 200 GB of RAM. The DRIVE dataset in the dbGap database was composed of 49,111 subjects genotyped for 528,620 SNPs using OncoArray (Amos et al., 2017). 55.4% of the subjects were from North America, 43.3% from Europe, and 1.3% from Africa. The disease outcome of the subjects was labeled as malignant tumor (48%), in situ tumor (5%), and no tumor (47%). In this study, the subjects in the malignant tumor and *in situ* tumor categories were labeled as cases and the subjects in the no tumor category were labeled as controls, resulting in 26053 (53%) cases and 23058 (47%) controls. The subjects in the case and control classes were randomly assigned to a training set (80%), a validation set (10%), and a test set (10%) (Figure 2.1). The association analysis was conducted on the training set using PLINK 2.0 (Chang et al., 2015). The p-value for each SNP was calculated using logistic regression.

2.2.2 Development of deep neural network models for PRS estimation

A variety of deep neural network (DNN) architectures (Bengio et al., 2009) were trained using Tensorflow 1.13. The Leaky Rectified Linear Unit (ReLU) activation function (Xu et al., 2015) was used on all hidden-layers neurons with the negative slope co-efficient set to 0.2. The output neuron used a sigmoid activation function. The training error was computed using the cross-entropy function:

$$\sum_{i=1}^{n} y * \log(p) + (1 - y) * \log(1 - p) \tag{2.1}$$

where $p \in [0, 1]$ is the prediction probability from the model and $y \in [|0, 1|]$ is the prediction target at 1 for case and 0 for control. DNNs were trained using mini-batches with a batch size of 512. The Adam optimizer (Kingma and Ba, 2014b), an

adaptive learning rate optimization algorithm, was used to update the weights in each mini-batch. The initial learning rate was set to 10^{-4}, and the models were trained for up to 200 epochs with early stopping based on the validation AUC score. Dropout (Srivastava et al., 2014) was used to reduce overfitting. Batch normalization (BN) (Ioffe and Szegedy, 2015) was used to accelerate the training process, and the momentum for the moving average was set to 0.9 in BN.

2.2.3 Development of alternative machine learning models for PRS estimation

Logistic regression, decision tree, random forest, AdaBoost, gradient boosting, support vector machine (SVM), and Gaussian naive Bayes were implemented and tested using the scikit-learn machine learning library in Python. These models were trained using the same training set as the DNNs and, similarly, their hyperparameters were tuned using the same validation set (Figure 2.1). These models are briefly described below.

- Decision Tree: The gini information gain with best split was used. The maximum depth was not set. The tree expanded until all leaves were pure or contained less than a minimum number of two examples per split.

- Random Forest: 3000 decision trees (as configured above) were used as base learners. Bootstrap samples were used to build each base learner. When searching for each tree's best split, the maximum number of considered features was set to be the square root of the number of features.

- AdaBoost: 2000 decision trees (as configured above) were used as base learners. The learning rate was set to 1. The algorithm used was SAMME.R (Hastie et al., 2009).

- Gradient Boosting: 400 decision trees (as configured above) were used as the base learners. Log-loss was used as the loss function. The learning rate was fixed to 0.1. The mean squared error with improvement score (Friedman, 2001) was used to measure the quality of a split.

- SVM: The kernel was a radial basis function with $\gamma = \frac{1}{n*Var}$, where n is the number of SNPs and Var is the variance of the SNPs across individuals. The regularization parameter C was set to 1.

- Logistic Regression: L2 regularization with $\alpha = 0.5$ was used. L1 regularization was tested, but not used, because it did not improve the performance.

- Gaussian Naïve Bayes: The likelihood of the features was assumed to be Gaussian. The classes had uninformative priors.

2.2.4 Development of statistical models for PRS estimation

The same training and validation sets were used to develop statistical models (Figure 2.1). The BLUP and BayesA models were constructed using the bWGR R package. The LDpred model was constructed using the algorithm as described (Vilhjálmsson et al., 2015).

- BLUP: The linear mixed model was $y = \mu + Xb + e$, where y were the response variables, μ were the intercepts, X were the input features, b were the regression coefficients, and e were the residual coefficients.

- BayesA: The priors were assigned from a mixture of normal distributions.

- LDpred: The p-values were generated by our association analysis described above. The validation set was provided as reference for LDpred data coordination. The

radius of the Gibbs sampler was set to be the number of SNPs divided by 3000 as recommended by the LDpred user manual (`https://github.com/bvilhjal/ldpred/blob/master/ldpred/run.py`).

The score distributions of DNN, BayesA, BLUP, and LDpred were analyzed with the Shapiro test for normality and the Bayesian Gaussian mixture (BGM) expectation maximization algorithm. The BGM algorithm decomposed a mixture of two Gaussian distributions with weight priors at 50

2.2.5 DNN model interpretation protocol

LIME and DeepLift were used to interpret the DNN predictions for subjects in the test set with DNN output scores higher than 0.67, which corresponded to a precision of 90%. For LIME, the submodular pick algorithm was used, the kernel size was set to 40, and the number of explainable features was set to 41. For DeepLift, the importance of each SNP was computed as the average across all individuals, and the reference activation value for a neuron was determined by the average value of all activations triggered across all subjects.

2.3 Results and Discussion

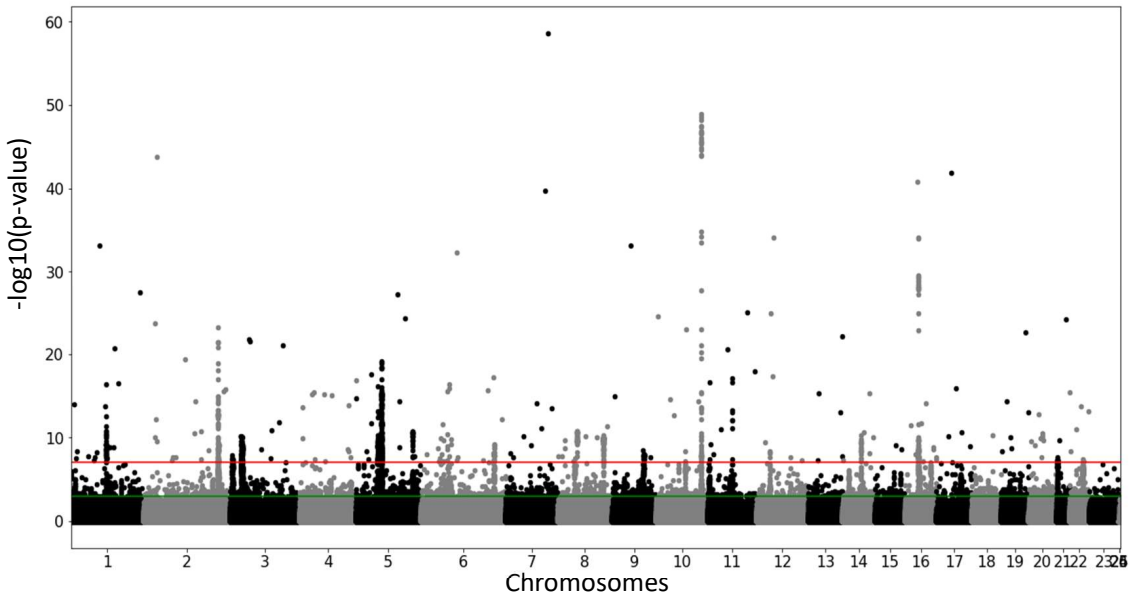
In this section, we discuss our results and provide rationals for why DNN performs better than the baseline algorithms to estimate Breast Cancer PRS.

2.3.1 Development of a machine learning model for breast cancer PRS estimation

The breast cancer GWAS dataset containing 26053 cases and 23058 controls was generated by the Discovery, Biology, and Risk of Inherited Variants in Breast Cancer (DRIVE) project (Amos et al., 2017). The DRIVE data is available from the NIH dbGaP database under the accession number of phs001265.v1.p1. The cases and controls were randomly split into a training set, a validation set, and a test set (Figure 2.1). The training set was used to estimate the p-values of SNPs using association analysis and train machine learning and statistical models. The hyperparameters of the machine learning and statistical models were optimized using the validation set. The test set was used for the final performance evaluation and model interpretation.

The statistical significance of the disease association with 528,620 SNPs was assessed with Plink using only the training set. The obtained p-values for all tested SNPs are shown in Figure 2.2A as a Manhattan plot.To obtain unbiased benchmarking results on the test set, it was critical not to use the test set in the association analysis (Figure 2.1) and not to use association p values from previous GWAS studies that included subjects in the test set, as well-described in the Section 7.10.2 of (Hastie et al., 2009). There were 1,061 SNPs with a p-value less than the critical value of $9.5 * 10^{-8}$, which was set using the Bonferroni correction ($9.5 * 10^{-8} = 0.05/528,620$). Filtering with a Bonferroni-corrected critical value may remove many informative SNPs that have small effects on the phenotype, epistatic interactions with other SNPs, or non-linear association with the phenotype (De et al., 2014). Relaxed filtering with higher p-value cutoffs was tested to find the optimal feature set for DNN (Figure 2.2B). The DNN models in Figure 2.2B had a deep feedforward architecture consisting of an input layer of variable sizes, followed by 3 successive hidden layers containing 1000,

(A)



(B)

p-value cutoff	SNPs	Computational Cost of Training		AUC		Accuracy	
		Convergence time (minutes)	Peak Memory (GB)	Training	Validation	Training	Validation
None	528,620	1308	66.6	100.0%	65.9%	100.0%	60.1%
10^{-2}	13,890	51	3.2	93.4%	66.5%	85.1%	61.4%
10^{-3}	5,273	23	2.2	80.5%	67.1%	73.4%	62.0%
10^{-4}	3,041	16	2	75.9%	66.4%	67.6%	61.1%
10^{-5}	2,099	9	1.4	72.2%	65.7%	63.2%	60.8%

Figure 2.2: **SNP filtering and model training for DNN.** (A) Manhattan plot from the association analysis. Each point represents a SNP with its p-value in the log10 scale on the y-axis and its position in a chromosome on the x-axis. The x-axis is labeled with the chromosome numbers. Chromosome 23 represents the X chromosome. Chromosomes 24 and 25 represent the pseudoautosomal region and non-pseudoautosomal region of the Y chromosome, respectively. Chromosome 26 designates the mitochondrial chromosome. The red line marks the p-value cutoff at $9.5 * 10^{-8}$and the green line marks the p-value cutoff at 10^{-3}. B) Performance of the DNN models trained using five SNP sets filtered with increasing p-value cutoffs. The models were compared by their training costs and performances in the training and validation sets.

250, and 50 neurons and an output layer with a single neuron. As the p-value cutoff increased, a greater number of SNPs were incorporated as input features, and training consumed a larger amount of computational resources in terms of computing time and peak memory usage. A feature set containing 5,273 SNPs above the p-value cutoff of 10^{-3} provided the best prediction performance measured by the AUC and accuracy on the validation set. In comparison with smaller feature sets from more stringent p-value filtering, the 5,273-SNP feature set may have included many informative SNPs providing additional signals to be captured by DNN for prediction. On the other hand, more relaxed filtering with p-value cutoffs greater than 10^{-3} led to significant overfitting as indicated by an increasing prediction performance in the training set and a decreasing performance in the validation set (Figure 2.2B).

Interestingly, the largest DNN model, consisting of all 528,620 SNPs, decreased the validation AUC score by only 1.2% and the validation accuracy by 1.9% from the highest achieved values. This large DNN model used an 80% dropout rate to obtain strong regularization, while all the other DNN models utilized a 50% dropout rate. This suggested that DNN was able to perform feature selection without using p-values, although the limited training data and the large neural network size resulted in complete overfitting. The effects of dropout and batch normalization were tested using the 5,273-SNP DNN model (Figure 2.3). Without dropout, the DNN model using only batch normalization had a 3.0% drop in AUC and a 4.0% drop in accuracy and its training converged in only two epochs. Without batch normalization, the DNN model had 0.1% higher AUC and 0.3% lower accuracy but its training required a 73% increase in the number of epochs to reach convergence.

As an alternative to filtering, autoencoding was tested to reduce a large number of SNPs to a small set of features for PRS estimation (Fergus et al., 2018; Cudic et al.,

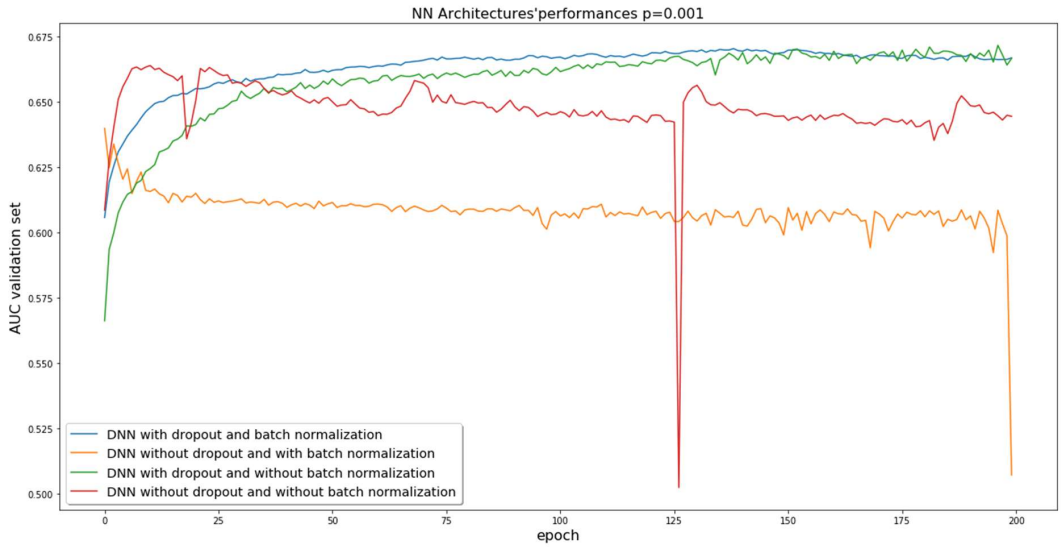


Figure 2.3: **Effects of dropout and batch normalization on the 5,273-SNP DNN model.**

2018). An autoencoder was trained to encode 5273 SNPs into 2000 features with a mean square error (MSE) of 0.053 and a root mean square error (RMSE) of 0.23. The encodings were used to train a DNN model with the same architecture as the ones shown in Figure 2.2B except for the number of input neurons. The autoencoder-DNN model had a similar number of input neurons for DNN as the 2099-SNP DNN model, but had a 1.3% higher validation AUC and a 0.2% higher validation accuracy (Table 2.1 and Figure 2.2B). This increased validation AUC and accuracy suggested that the autoencoding provided improved dimensionality reduction compared to the SNP filtering based on p-values. In comparison with the 5,273-SNP model, auto-encoding sped up the convergence process, but led to a 0.3% reduction in validation AUC score and a 1.6% reduction in validation accuracy.

The deep feedforward architecture benchmarked in Figure 2.2B was compared with a number of alternative neural network architectures using the 5,273-SNP feature set (Table 2.1). A shallow neural network with only one hidden layer resulted in a 0.9% lower AUC and 1.1% lower accuracy in the validation set compared to the DNN. This suggested that additional hidden layers in DNN were useful in representing complex interactions among SNPs. The additional hidden layers also supported additional feature selection and transformation in the model. One-dimensional convolutional neural network (1D CNN) was previously used to estimate the PRS for bone heel mineral density, body mass index, systolic blood pressure and waist-hip ratio (Bellot et al., 2018) and was also tested here for breast cancer prediction with the DRIVE dataset. The validation AUC and accuracy of 1D CNN were lower than DNN by 3.2% and 2.0%, respectively. Two-dimensional CNN was particularly popular for image analysis, because the receptive field of the convolutional layer can capture space-invariant information with shared parameters. However, the SNPs distributed across a genome

Model	Architecture	Validation AUC	Validation Accuracy	Convergence (# Epochs)
DNN	3 hidden layers with 1000, 250, and 50 neurons. Dropout and batch normalization (BN) enabled	67.1%	62.0%	110
Shallow NN (SNN)	1 hidden layer with 50 neurons. With dropout but no BN	66.2%	60.9%	20
1D Convolutional NN (1D CNN)	2 convolution layers with max pooling followed by 3 hidden layers with 1000, 250, and 50 neurons. Dropout and BN enabled	63.9%	59.9%	155
Autoencoder-DNN	autoencoding with no hidden layer followed by DNN with dropout and BN enabled	67.0%	61.0%	31

Table 2.1: **Effects of dropout and batch normalization on the 5,273-SNP DNN model.**

may not have significant space-invariant patterns to be captured by the convolutional layer, which may explain the poor performance of CNN.

The 5,273-SNP feature set was used to test alternative machine learning approaches, including logistic regression, decision tree, naive Bayes, random forest, ADAboost, gradient boosting, and SVM, for PRS estimation (Figure 2.4). These models were trained, turned, and benchmarked using the same training, validation, and test sets, respectively, as the DNN models (Figure 2.1). Although the decision tree had a test AUC of only 50.9%, ensemble algorithms that used decision trees as the base learner, including random forest, ADABoost, and gradient boosting, reached test AUCs of 63.6%, 64.4%, and 65.1%, respectively. This showed the advantage of ensemble learning. SVM reached a test AUC of 65.6%. Naïve Bayes and logistic regression were both linear models with the assumption of independent features. Logistic regression performed substantially better than naïve Bayes, ensemble techniques and SVM, based on the AUC scores. Out of all the machine learning models, the DNN model still achieved the highest test AUC at 67.4% and the highest test accuracy at 62.8%.

2.3.2 Comparison of the DNN model with statistical models for breast cancer PRS estimation

The performance of DNN was compared with three representative statistical models, including BLUP, BayesA, and LDpred (Table 2.2). Because the relative performance of these methods may be dependent on the number of training examples available, the original training set containing 39,289 subjects was down-sampled to create three smaller training sets containing 10000, 20000, 30000 subjects. As the 5,273-SNP feature set generated with a p-value cutoff of 10^{-3} may not be the most appropriate for the statistical methods, a 13,890-SNP feature set (p-value cutoff = 10^{-2}) and a 2,099-SNP

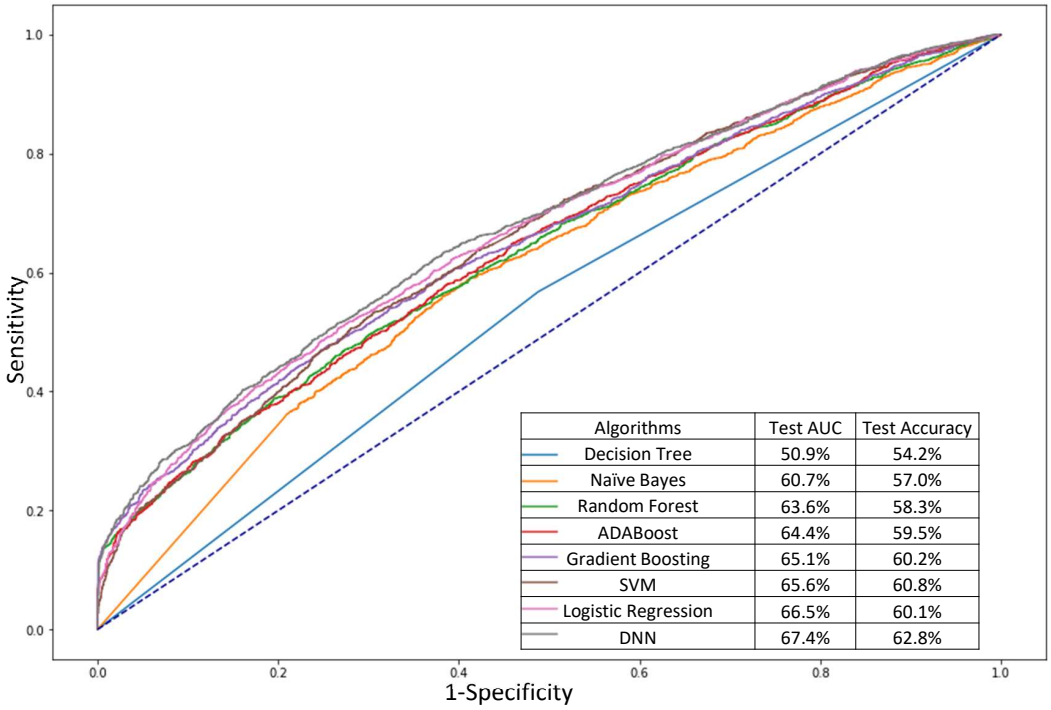


Figure 2.4: **Comparison of machine learning approaches for PRS estimation.** The performances of the models were represented as Receiver Operating Characteristic (ROC) curves in different colors. The Area under the ROC curve (AUC) and the accuracy from the test set are shown in the legend. The DNN model outperformed the other machine learning models in terms of AUC and accuracy.

Algorithms	DNN			BLUP			BayesA			LDpred		
PC* / TS**	10^{-5}	10^{-3}	10^{-2}	10^{-5}	10^{-3}	10^{-2}	10^{-5}	10^{-3}	10^{-2}	10^{-5}	10^{-3}	10^{-2}
10,000	64.8%	65.5%	65.1%	63.5%	62.5%	60.6%	63.7%	62.0%	59.9%	60.8%	62.4%	61.6%
20,000	65.6%	66.6%	66.4%	62.9%	63.0%	60.6%	62.7%	63.0%	60.4%	60.8%	62.4%	61.6%
30,000	66.0%	66.9%	66.6%	64.2%	63.1%	60.7%	64.3%	63.1%	60.7%	60.7%	62.4%	61.6%
39,289	66.2%	67.4%	67.3%	64.2%	63.3%	61.0%	64.5%	63.4%	61.1%	60.7%	62.4%	61.6%

*: p-value cutoff

**: training set size

Table 2.2: **AUC test scores of DNN, BLUP, BayesA, and LDpred models at different p-value cutoffs (PC) and training set sizes (TS).**

feature set (p-value cutoff = 10^{-5}) were tested for all methods. Although LDpred also required training data, its prediction relied primarily on the provided p-values, which were generated for all methods using all 39,289 subjects in the training set. Thus, the down-sampling of the training set did not reduce the performance of LDpred. LDpred reached its highest AUC score at 62.4% using the p-value cutoff of 10^{-3}. A previous study (Ge et al., 2019)[12] that applied LDpred to breast cancer prediction using the UK Biobank dataset similarly obtained an AUC score of 62.4% at the p-value cutoff of 10^{-3}. This showed consistent performance of LDpred in the two studies using different datasets. When DNN, BLUP, and BayesA used the full training set, they obtained higher AUCs than LDpred at their optimum p-value cutoffs. DNN, BLUP, and BayesA all gained performance with the increase in the training set sizes (Table 2.2). The performance gain was more substantial for DNN than BLUP and BayesA. The increase from 10,000 subjects to 39,258 subjects in the training set resulted in a 1.9% boost to DNN's best AUC, a 0.7% boost to BLUP, and a 0.8% boost to BayesA. This indicated the different variance-bias trade-offs made by DNN, BLUP, and BayesA. The high variance of DNN required more training data, but could capture more extensive interactions among SNPs and non-linear relationships between the SNPs and the phenotype. The high bias of BLUP and BayesA had lower risk for overfitting using smaller training sets, but their models only considered linear relationships. The higher AUCs of DNN across all training set sizes indicated that DNN had a better variance-bias balance for breast cancer PRS estimation. For all four training set sizes, BLUP and BayesA achieved higher AUCs using more stringent p-value filtering. When using the full training set, reducing the p-value cutoffs from 10-2 to 10-5 increased the AUCs of BLUP from 61.0% to 64.2% and the AUCs of BayesA from 61.1% to 64.5%. This suggested that BLUP and BayesA preferred a reduced number of SNPs that were found by logistic regression to be significantly associated with the phenotype. On the

other hand, DNN produced lower AUCs using the p-value cutoff of 10-5 than the other two higher cutoffs. This suggested that DNN can perform better feature selection in comparison to SNP filtering based on p-values from logistic regression.

The four algorithms were compared using the score histograms of the case population and the control population from the test set in Figure 2.5. The score distributions of BLUP, BayesA and LDpred all followed normal distributions. The p-values from the Shapiro normality test of the case and control distributions were 0.46 and 0.43 for BayesA, 0.50 and 0.95 for BLUP, and 0.17 and 0,24 for LDpred, respectively. The case and control distributions were $N_{case}(\mu = 0.577, \sigma = 0.20)$ and $N_{control}(\mu = 0.479, \sigma = 0.19)$ from BayesA, $N_{cases}(\mu = 0.572, \sigma = 0.19)$ and $N_{control}(\mu = 0.483, \sigma = 0.18)$ from BLUP, and $N_{case}(\mu = -33.52, \sigma = 5.4)$ and $N_{control}(\mu = -35.86, \sigma = 4.75)$ from LDpred. The means of the case distributions were all significantly higher than the control distributions for BayesA (p-value $< 10^{-16}$), BLUP (p-value $< 10^{-16}$), and LDpred (p-value $< 10^{-16}$) and their case and control distributions had similar standard deviations. The score histograms of DNN did not follow normal distributions based on the Shapiro normality test with a p-value of 4.1×10^{-34} for the case distribution and a p-value of 2.5×10^{-9} for the control distribution. The case distribution had the appearance of a bi-modal distribution. The Bayesian Gaussian mixture expectation maximization algorithm decomposed the case distribution to two normal distributions: $N_{case1}(\mu = 0.519, \sigma = 0.096)$ with an 86.5% weight and $N_{case2}(\mu = 0.876, \sigma = 0.065)$ with a 13.5% weight. The control distribution was resolved into two normal distributions with similar means and distinct standard deviations: $N_{control1}(\mu = 0.471, \sigma = 0.1)$ with an 85.0% weight and $N_{control2}(\mu = 0.507, \sigma = 0.03)$ with a 15.0% weight. The N_{case1} distribution had a similar mean as the $N_{control1}$ and $N_{control2}$ distributions. This suggested that the N_{case1}

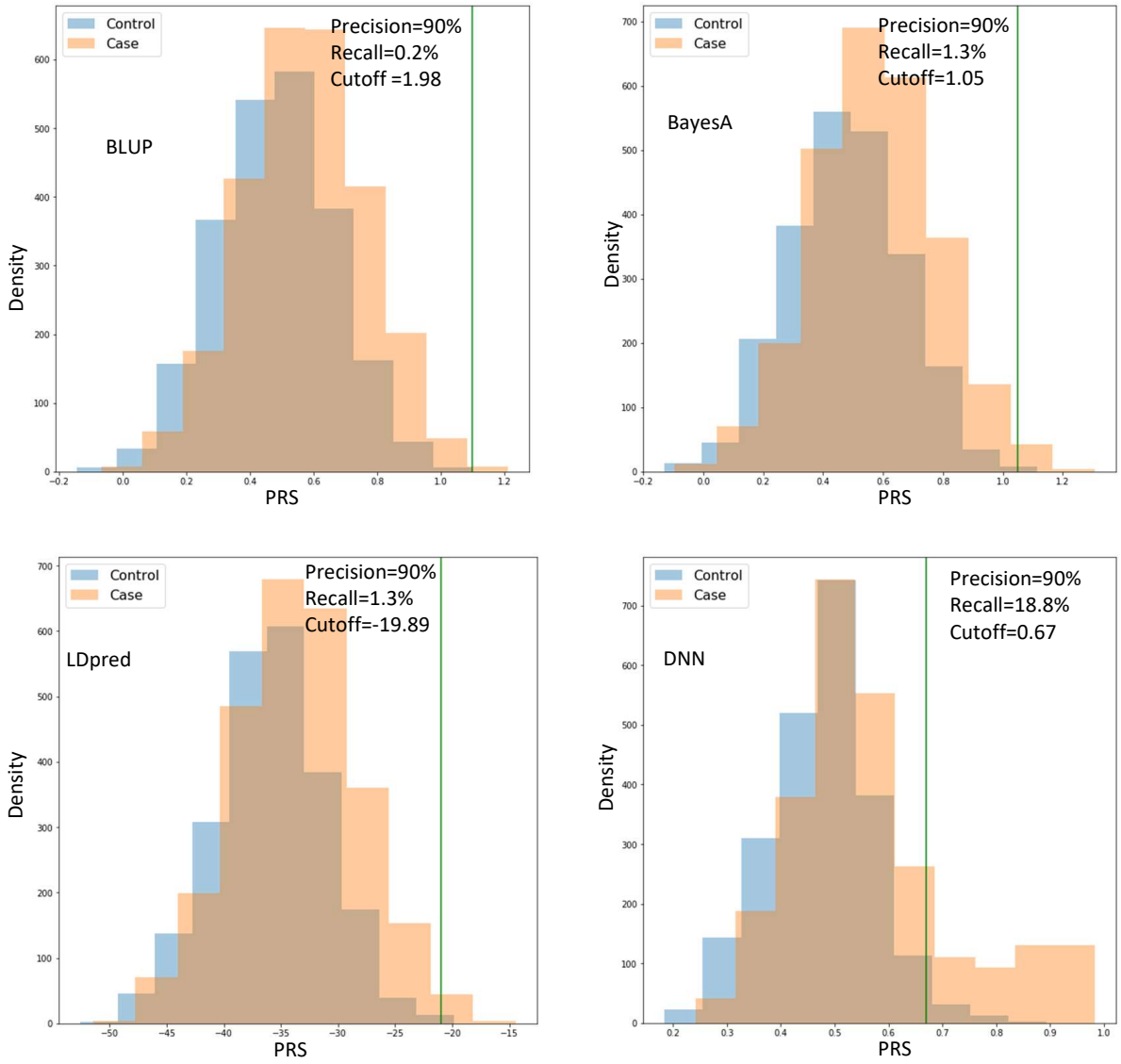


Figure 2.5: **Score histograms of DNN, BLUP, BayesA, and LDpred.** The case and control populations are shown in the orange and blue histograms, respectively. The green line represents the score cutoff corresponding to the precision of 90% for each model. DNN had a much higher recall than the other algorithms at 90% precision.

distribution may represent a normal-genetic-risk case sub-population, in which the subjects may have a normal level of genetic risk for breast cancer and the oncogenesis likely involved a significant environmental component. The mean of the N_{case2} distribution was higher than the means of both the N_{case1} and $N_{control1}$ distributions by more than 4 standard deviations (p-value $< 10^{-16}$). We hypothesized that the N_{case2} distribution represented a high-genetic-risk case sub-population for breast cancer, in which the subjects may have inherited many genetic variations associated with breast cancer.

Three GWAS were performed between the high-genetic risk case subpopulation with DNN PRS > 0.67, the normal genetic-risk case subpopulation with DNN PRS < 0.67, and the control population (see Supplementary Table 3 in Badré et al. (2021)). The GWAS analysis of the high-genetic-risk case subpopulation versus the control population identified 182 significant SNPs at the Bonferroni level of statistical significance. The GWAS analysis of the high-genetic-risk case subpopulation versus the normal-genetic-risk case subpopulation identified 216 significant SNPs. The two sets of significant SNPs found by these two GWAS analyses were very similar, sharing 149 significant SNPs in their intersection. Genes associated with these 149 SNPs were investigated with pathway enrichment analysis (Fisher's Exact Test; $P < 0.05$) using SNPnexus (Dayem Ullah et al., 2018) (see Supplementary Table 4 in (Badré et al., 2021)). Many of the significant pathways were involved in DNA repair (O'Connor, 2015) signal transduction (Kolch et al., 2015), and suppression of apoptosis (Fernald and Kurokawa, 2013). Interestingly, the GWAS analysis of the normal genetic-risk case subpopulation and the control population identified no significant SNP. This supported our classification of the cases into the normal-genetic-risk subjects, and Deep neural network improves the estimation of polygenic risk scores for breast cancer 365 the high-genetic-risk subjects based on their PRS scores from the DNN model.

In comparison with AUCs, it may be more relevant for practical applications of PRS to compare the recalls of different algorithms at a given precision that warrants clinical recommendations. At 90% precision, the recalls were 18.8% for DNN, 0.2% for BLUP, 1.3% for BayesA, and 1.3% for LDpred in the test set of the DRIVE cohort with a 50% prevalence. This indicated that DNN can make a positive prediction for 18.8% of the subjects in the DRIVE cohort and these positive subjects would have an average chance of 90% to eventually develop breast cancer. However, BLUP, BayesA and LDpred can only make a similarly confident prediction for less than 2% of the subjects. American Cancer Society advises yearly breast MRI and mammogram starting at the age of 30 years for women with a lifetime risk of breast cancer greater than 20%, which meant a 20% precision for PRS. By extrapolating the performance in the DRIVE cohort, the DNN model should be able to achieve a recall of 65.4% at a precision of 20% in the general population with a 12% prevalence rate of breast cancer.

2.3.3 Interpretation of the DNN model

While the DNN model used 5,273 SNPs as input, we hypothesized that only a small set of these SNPs were particularly informative for identifying the subjects with high genetic risks for breast cancer. LIME and DeepLift were used to find the top-100 salient SNPs used by the DNN model to identify the subjects with classification scores higher than the cutoff at 90% precision. 23 SNPs were ranked by both algorithms to be among their top-100 salient SNPs (Figure 2.6). The small overlap between their results can be attributed to their different interpretation approaches. LIME considered the DNN model as a black box and perturbed the input to estimate the importance of each variable; whereas, DeepLift analyzed the gradient information of the DNN model. 30% of LIME's salient SNPs and 49% of DeepLift's salient SNPs had p-values less

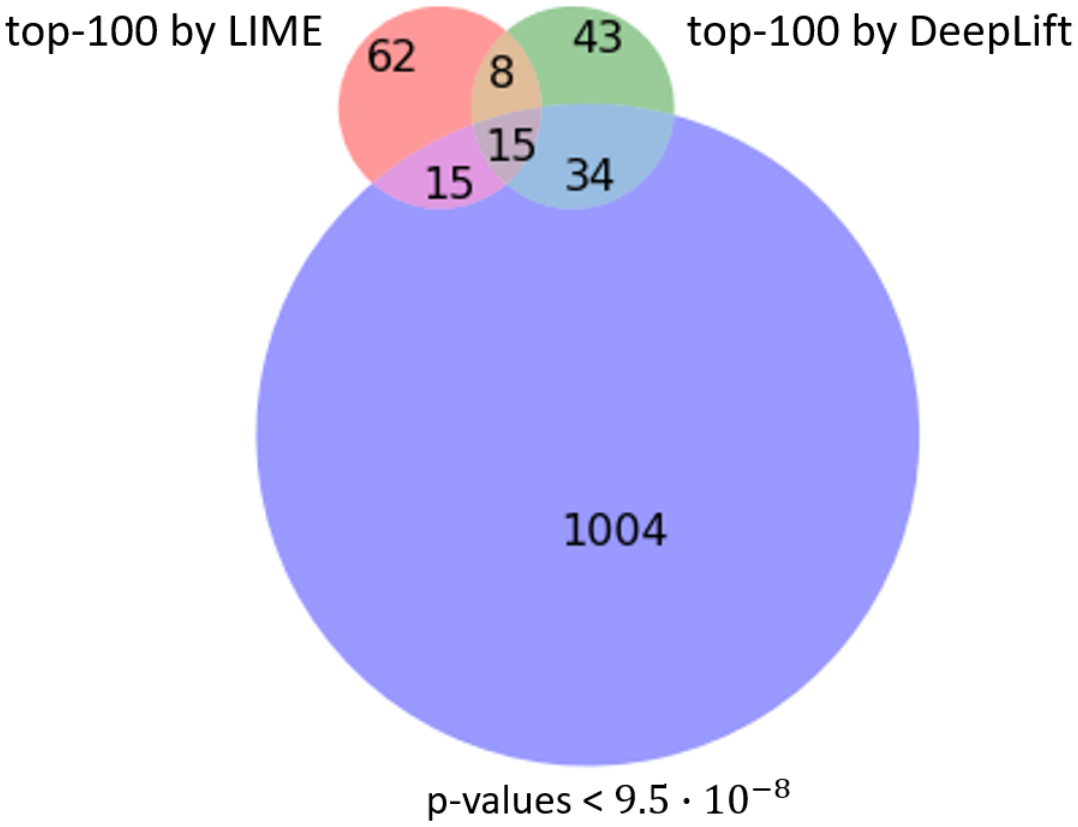


Figure 2.6: **Venn diagram of important SNPs found by LIME, DeepLift, and association analysis.** The red circle represents the top-100 salient SNPs identified by LIME. The green circle represents the top-100 salient SNPs identified by DeepLift. The blue circle represents the 1,068 SNPs that had p-values lower than the Bonferroni-corrected critical value. The numbers in the Venn diagram show the sizes of the intersections and complements among the three sets of SNPs.

SNP	locus	LIME (10^{-4})	DeepLift (10^{-2})	p-value	MAF*	Genes of interest**
corect_rs139337779	12q24.22	4.5	-3.3	6.5E-04	11%	NOS1
chr13_113796587_A_G	13q34	4.3	-3.8	2.8E-04	3%	F10
chr9_16917672_G_T	9p22.2	4.5	-2.5	7.6E-05	4%	BNC2/RP11-132E11.2
chr8_89514784_A_G	8q21.3	27.0	3.7	2.5E-05	56%	RP11-586K2.1
chr17_4961271_G_T	17p13.2	4.2	-2.2	8.2E-06	4%	SLC52A1/RP11-46I8.1
rs11642757	16q23.2	5.3	-2.9	2.0E-06	6%	RP11-345M22.1
rs4040605	1p36.33	4.4	2.4	9.6E-07	37%	RP11-54O7.3/SAMD11
chr5_180405432_G_T	5q35.3	4.1	-3.4	2.3E-07	3%	CTD-2593A12.3/CTD-2593A12.4
Chr6:30954121:G:T	6p21.33	6.8	4.9	1.0E-08	42%	MUC21
chr14_101121371_G_T	14q32.2	5.8	3.9	1.0E-10	33%	CTD-2644I21.1/LINC00523
rs12542492	8q21.11	40.0	2.8	6.3E-11	34%	RP11-434I12.2
corect_rs116995945	17q22	3.6	-4.5	2.5E-11	5%	SCPEP1/RNF126P1
chr14_76886176_C_T	14q24.3	3.5	2.3	2.3E-11	41%	ESRRB
chr2_171708059_C_T	2q31.1	4.1	-6.7	1.9E-11	7%	GAD1
chr7_102368966_A_G	7q22.1	4.1	-2.6	6.8E-12	4%	RASA4DP/FAM185A
chr8_130380476_C_T	8q24.21	4.3	2.5	4.7E-12	22%	CCDC26
corect_rs181578054	22q13.33	4.1	3.0	7.1E-14	40%	ARSA/Y_RNA
rs3858522	11p15.5	7.7	3.3	2.2E-17	52%	H19/IGF2
chr3_46742523_A_C	3p21.31	5.2	4.9	1.8E-22	35%	ALS2CL/TMIE
chr13_113284191_C_T	13q34	4.0	-4.0	7.8E-23	5%	TUBGCP3/C13orf35
chr1_97788840_A_G	1p21.3	6.0	-6.8	6.6E-34	9%	DPYD
chr7_118831547_C_T	7q31.31	4.0	-3.5	1.9E-40	6%	RP11-500M10.1/AC091320.2
chr16_52328666_C_T	16q12.1	23.0	5.2	1.5E-41	21%	RP11-142G1.2/TOX3

*Minor Allele Frequency

** < 300kb from target SNPs

Table 2.3: **Top salient SNPs identified by both LIME and DeepLift from the DNN model**

than the Bonferroni significance threshold of 9.5×10^{-8}. This could be attributed to the non-linear relationship between the salient SNPs and the disease outcome, which cannot be captured by association analysis using logistic regression.

To illustrate this, four salient SNPs with significant p-values were shown in Figure 2.7, which exhibited linear relationships between their genotype values and log odds ratios as expected. Four salient SNPs with insignificant p-values were shown in Figure 2.8, which showed clear biases towards cases or controls by one of the genotype values in a nonlinear fashion.

Michailidou et al. (2017) summarized a total of 172 SNPs associated with breast cancer. Out of these SNPs, 59 were not included on OncoArray, 63 had an association p value less than 103 and were not included in the 5273-SNP feature set for DNN, 34 were not ranked among the top-1000 SNPs by either DeepLIFT or LIME, and 16 were ranked among the top-1000 SNPs by DeepLIFT, LIME, or both (see Supplementary Table 5 in (Badré et al., 2021)). This indicates that many SNPs with significant association may be missed by the interpretation of DNN models.

The 23 salient SNPs identified by both DeepLift and LIME in their top-100 list are shown in Table 2.3. Eight of these SNPs had p-values higher than the Bonferroni level of significance and were missed by the association analysis using Plink. The potential oncogenesis mechanisms for some of the eight SNPs have been investigated in previous studies. The SNP, rs139337779 at 12q24.22, is located within the gene, Nitric oxide synthase 1 (NOS1). (Li et al., 2019) showed that the overexpression of NOS1 can upregulate the expression of ATP-binding cassette, subfamily G, member 2 (ABCG2), which is a breast cancer resistant protein (Mao and Unadkat, 2015), and NOS1-indeuced chemo-resistance was partly mediated by the upregulation of ABCG2 expression. (Lee et al., 2009) reported that NOS1 is associated with the breast cancer

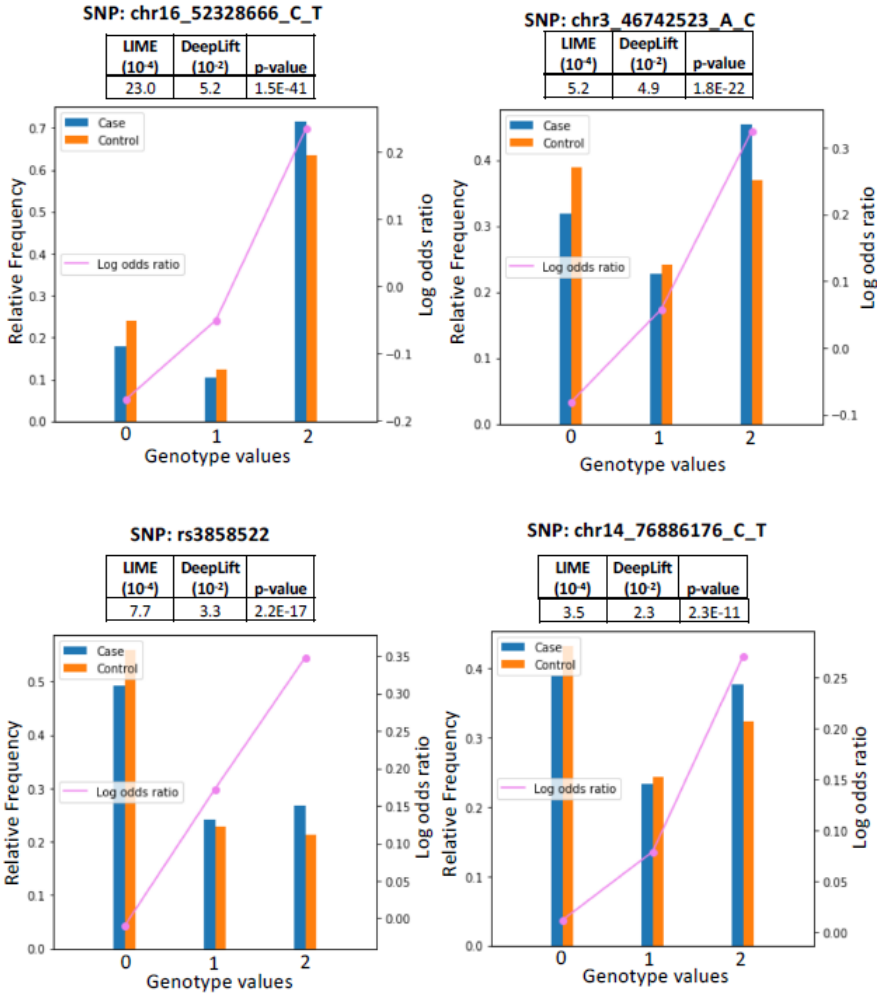


Figure 2.7: **Genotype-phenotype relationships for salient SNPs used in the DNN model: Linear case** Four salient SNPs with linear relationships as shown by the pink lines and the significant association p-values.

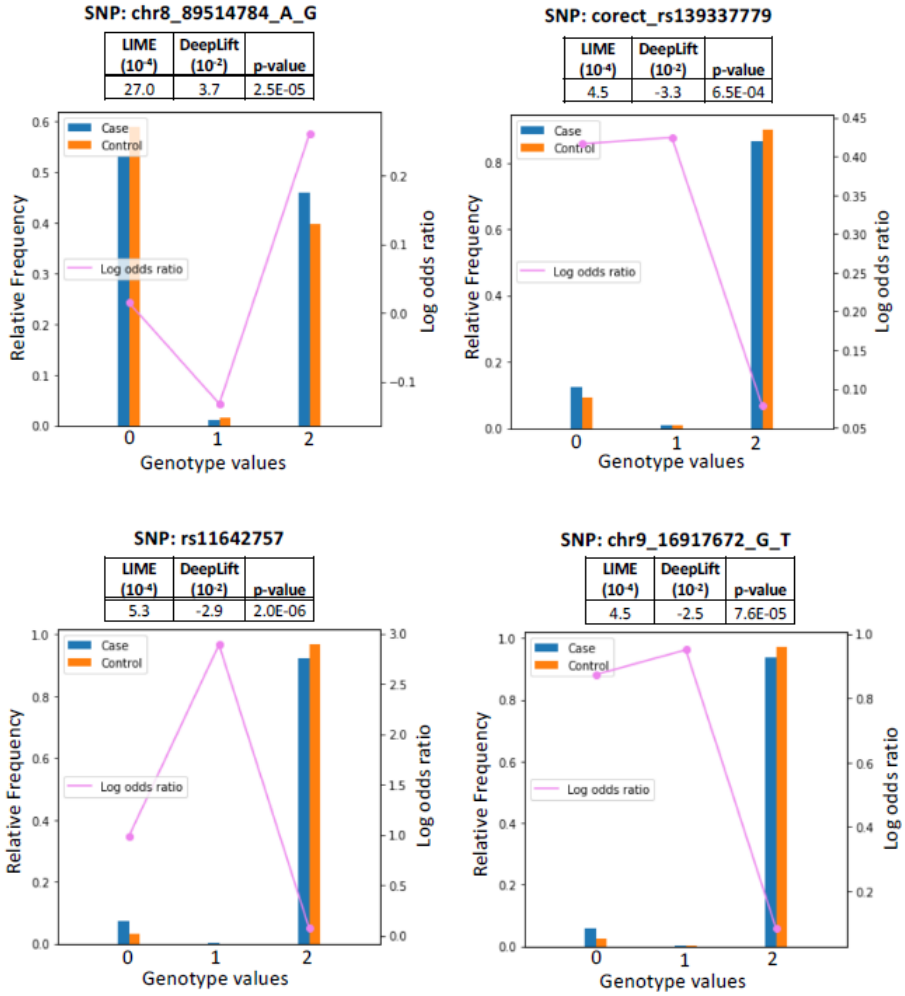


Figure 2.8: **Genotype-phenotype relationships for salient SNPs used in the DNN model: Non-linear case.** Four salient SNPs with non-linear relationships as shown by the pink lines and the insignificant association p-values. The DNN model was able to use SNPs with non-linear relationships as salient features for prediction.

risk in a Korean cohort. The SNP, chr13_113796587_A_G at 13q34, is located in the F10 gene, which is the coagulation factor (Tinholt et al., 2014) showed that the increased coagulation activity and genetic polymorphisms in the F10 gene are associated with breast cancer. The BNC2 gene containing the SNP, chr9_16917672_G_T at 9p22.2, is a putative tumor suppressor gene in high-grade serious ovarian carcinoma (Cesaratto et al., 2016). The SNP, chr2_171708059_C_T at 2q31.1, is within the GAD1 gene and the expression level of GAD1 is a significant prognostic factor in lung adenocarcinoma (Tsuboi et al., 2019). Thus, the interpretation of DNN models may identify novel SNPs with nonlinear association with the breast cancer (Purcell Shaun et al., 2009; Scott et al., 2017; LeBlanc and Kooperberg, 2010; Angermueller et al., 2016; Schmidhuber, 2015).

Chapter 3

LINA: A Linearizing Neural Network Architecture for Accurate First-Order and Second-Order Interpretations

3.1 Introduction

An interpretable machine learning algorithm should have a high representational capacity to provide strong predictive performance, and its learned representations should be amenable to model interpretation and understandable to humans. The two desiderata are generally difficult to balance. Linear models and decision trees generate simple representations for model interpretation but have low representational capacities for only simple prediction tasks. Neural networks and support vector machines have high representational capacities to handle complex prediction tasks, but their learned representations are often considered to be "black boxes" for model interpretation (Bermeitinger. et al., 2019). Predictive genomics is an exemplary application that requires both a strong predictive performance and high interpretability. In this application, the genotype information for a large number of SNPs in a subject's genome is used to predict the phenotype of this subject. While neural networks have been shown to provide better predictive performance than statistical models (Badré et al., 2021; Fergus et al., 2018), statistical models are still the dominant methods for predictive genomics, because geneticists and genetic counselors can understand which SNPs are used and how they are used as the basis for certain phenotype predictions. Neural network models have also been used in many other important bioinformatics applications (Ho Thanh Lam et al., 2020; Do and Le, 2020; Baltres et al., 2020) that can benefit from model interpretation. To make neural networks more useful for predictive genomics and other applications, we developed a new neural network architecture, referred to as linearizing neural network architecture (LINA), to provide both first-order and second-order interpretations and both instance-wise and model-wise interpretations. Model interpretation reveals the input-to-output relationships that a machine learning model has learned from the training data to make predictions (Molnar, 2020). The first-order model interpretation

aims to identify individual features that are important for a model to make predictions. For predictive genomics, this can reveal which individual SNPs are important for phenotype prediction. The second-order model interpretation aims to identify important interactions among features that have a large impact on model prediction. The second-order interpretation may reveal the XOR interaction between the two features that jointly determine the output. For predictive genomics, this may uncover epistatic interactions between pairs of SNPs (Cordell, 2002; Phillips, 2008). A general strategy for the first-order interpretation of neural networks, first introduced by Saliency (Simonyan et al., 2014), is based on the gradient of the output with respect to (w.r.t.) the input feature vector. A feature with a larger partial derivative of the output is considered more important. The gradient of a neural network model w.r.t. the input feature vector of a specific instance can be computed using backpropagation, which generates an instance-wise first-order interpretation. The Grad*Input algorithm (Shrikumar et al., 2017) multiplies the obtained gradient element-wise with the input feature vector to generate better scaled importance scores. As an alternative to using the gradient information, the Deep Learning Important FeaTures (DeepLIFT) algorithm explains the predictions of a neural network by backpropagating the activations of the neurons to the input features (Shrikumar et al., 2017). The feature importance scores are calculated by comparing the activations of the neurons with their references, which allows the importance information to pass through a zero gradient during backpropagation. The Class Model Visualization (CMV) algorithm (Simonyan et al., 2014) computes the visual importance of pixels in convolution neural network (CNN). It performs backpropagation on an initially dark image to find the pixels that maximize the classification score of a given class. While the algorithms described above were developed specifically for neural networks, model-agnostic interpretation algorithms can be used for all types of machine learning models. Local Interpretable Model-agnostic

Explanations (LIME) (Ribeiro et al., 2016) fits a linear model to synthetic instances that have randomly perturbed features in the vicinity of an instance. The obtained linear model is analyzed as a local surrogate of the original model to identify the important features for the prediction on this instance. Because this approach does not rely on gradient computation, LIME can be applied to any machine learning model, including non-differentiable models. Previously, we combined LIME and DeepLIFT to interpret a feedforward neural network model for predictive genomics (Badré et al., 2021). Kernel SHapley Additive exPlanations (SHAP) (Lundberg and Lee, 2017) uses a sampling method to find the Shapley value for each feature of a given input. The Multi-Objective Counterfactuals (MOC) method (Dandl et al., 2020) searches for the counterfactual explanations for an instance by solving a multi-objective optimization problem. The importance scores calculated by the L2X algorithm (Chen et al., 2018) are based on the mutual information between the features and the output from a machine learning model. L2X is efficient because it approximates the mutual information using a variational approach. The second-order interpretation is more challenging than the first-order interpretation because d features would have $\frac{d^2-d}{2}$ possible interactions to be evaluated. Computing the Hessian matrix of a model for the second-order interpretation is conceptually equivalent to, but much more computationally expensive than, computing the gradient for the first-order interpretation. Group Expected Hessian (GEH) (Cui et al., 2019) computes the Hessian of a Bayesian neural network for many regions in the input feature space and aggregates them to estimate an interaction score for every pair of features. The additive grooves algorithm (Sorokina et al., 2007) estimates the feature interaction scores by comparing the predictive performance of the decision tree containing all features with that of the decision trees with pairs of features removed. Neural Interaction Detection (NID) (Tsang et al., 2017) avoids the high computational cost of evaluating every feature pair by directly analyzing the weights

in a feedforward neural network. If some features are strongly connected to a neuron in the first hidden layer and the paths from that neuron to the output have high aggregated weights, then NID considers these features to have strong interactions. Model interpretations can be further classified as instance-wise interpretations or model-wise interpretations. Instance-wise interpretation algorithms, including Saliency (Simonyan et al., 2014), LIME (Ribeiro et al., 2016) and L2X (Chen et al., 2018), provide an explanation for a model's prediction for a specific instance. For example, an instance-wise interpretation of a neural network model for predictive genomics may highlight the important SNPs in a specific subject which are the basis for the phenotype prediction of this subject. This is useful for intuitively assessing how well grounded the prediction of a model is for a specific subject. Model-wise interpretation provides insights into how a model makes predictions in general. CMV (Simonyan et al., 2014) was developed to interpret CNN models. Instance-wise interpretation methods can also be used to explain a model by averaging the explanations of all the instances in a test set. A model-wise interpretation of a predictive genomics model can reveal the important SNPs for a phenotype prediction in a large cohort of subjects. Model-wise interpretations shed light on the internal mechanisms of a machine learning model. In this study, we designed the LINA architecture and developed the first-order and second-order interpretation algorithms for LINA. The interpretation performance of the new methods was benchmarked using synthetic datasets and a predictive genomics application in comparison with state-of-the-art (SOTA) interpretation methods. The interpretations from LINA were more versatile and more accurate than those from the SOTA methods.

3.2 Methods

3.2.1 LINA Architecture

The key feature of the LINA architecture (Figure 3.1) is the linearization layer, which computes the output as an element-wise multiplication product of the input features, attention weights, and coefficients:

$$y = S[K^T(A \circ X) + b] = S(\sum_{i=1}^{d}(k_i * a_i * x_i) + b) \tag{3.1}$$

where y is the output, X is the input feature vector, $S()$ is the activation function of the output layer, $\circ$ represents the element-wise multiplication operation, K and b are respectively the coefficient vector and bias that are constant for all instances, and A is the attention vector that adaptively scales the feature vector of an instance. X, A and K are three vectors of dimension d, which is the number of input features. The computation by the linearization layer and the output layer is also expressed in a scalar format in Equation 3.1. This formulation allows the LINA model to learn a linear function of the input feature vector, coefficient vector, and attention vector. The attention vector is computed from the input feature vector using a multi-layer neural network, referred to as the inner attention neural network in LINA. The inner attention neural network must be sufficiently deep for a prediction task owing to the designed low representational capacity of the remaining linearization layer in a LINA model. In the inner attention neural network, all hidden layers use a non-linear activation function, such as ReLU, but the attention layer uses a linear activation function to avoid any restriction in the range of the attention weights. This is different from the typical attention mechanism in existing attentional architectures which generally use the softmax activation function.

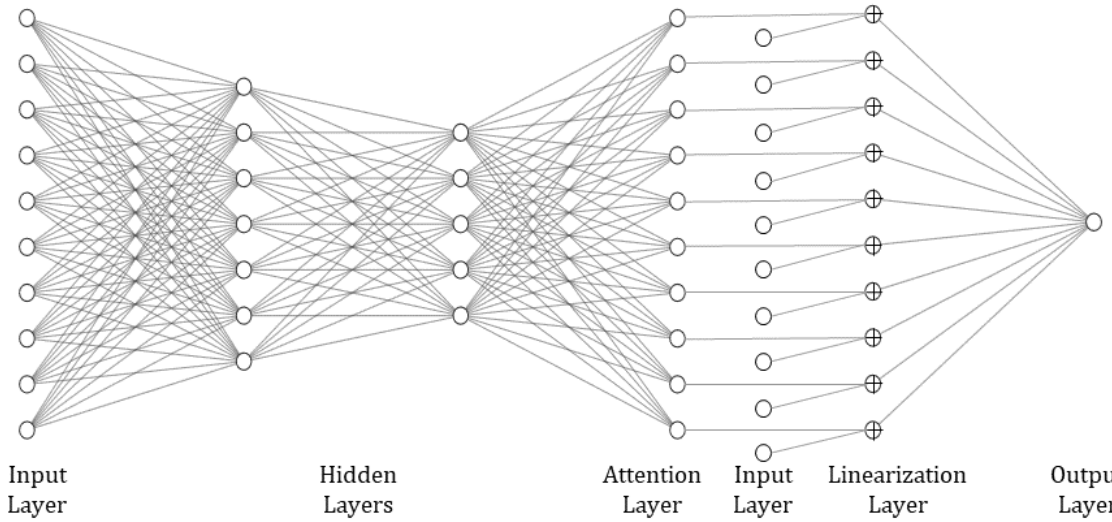


Figure 3.1: **An example of LINA model for structured data.** The LINA model uses an input layer and multiple hidden layer to output the attention weights in the attention layer. The attention weights are then multiplied with the input features element-wise in the linearization layer and then with the coefficients in the output layer. The crossed neurons in the linearization layer represent element-wise multiplication of their two inputs. The incoming connections to the crossed neurons have a constant weight of 1.

3.2.2 The Loss Function

The loss function for LINA is composed of the training error loss, regularization penalty on the coefficient vector, and regularization penalty on the attention vector:

$$loss = E(Y, Y_{true}) + \beta * ||K||_2 + \gamma * ||A - 1||_1 \tag{3.2}$$

where E is a differentiable convex training error function, $||K||_2$ is the L2 norm of the coefficient vector, $||A - 1||_1$ is the L1 norm of the attention vector minus 1, and β and γ are the regularization parameters. The coefficient regularization sets 0 to be the expected value of the prior distribution for K, which reflects the expectation of uninformative features. The attention regularization sets 1 to be the expected value of the prior distribution for A, which reflects the expectation of a neutral attention weight that does not scale the input feature. The values of β and γ and the choices of L2, L1, and L0 regularization for the coefficient and attention vectors are all hyperparameters that can be optimized for predictive performance on the validation set.

3.2.3 First-order interpretation

Interpretation from the gradient of the output, y, w.r.t the input feature vector, X. The output gradient can be decomposed as follows:

$$\frac{\partial y}{\partial x_i} = k_i * a_i + \sum_{j=1}^{d} \frac{\partial a_j}{\partial x_i} * x_j \tag{3.3}$$

Proof:

Let us derive $\frac{\partial y}{\partial x_i}$ for a **regression task** (or the logit for a classification task):

$$\frac{\partial y}{\partial x_i} = \frac{\partial k_i a_i x_i}{\partial x_i} + \sum_{j=1, j\neq i}^{d} \frac{\partial k_j a_j x_j}{\partial x_i} + \frac{\partial b}{\partial x_i}$$

$$= k_i \frac{\partial a_i x_i}{\partial x_i} + \sum_{j=1, j\neq i}^{d} k_j \frac{\partial a_j x_j}{\partial x_i}$$

$$= k_i (\frac{\partial a_i}{\partial x_i} + a_i) + \sum_{j=1, j\neq i}^{d} k_j \frac{\partial a_j x_j}{\partial x_i}$$

$$\frac{\partial y}{\partial x_i} = k_i a_i + \sum_{j=1}^{d} \frac{\partial a_j}{\partial x_i} x_j$$

End-of-proof.

The decomposition of the output gradient in LINA shows that the contribution of a feature in an attentional architecture comprises (i) a direct contribution to the output weighted by its attention weight and (ii) an indirect contribution to the output during attention computation. This indicates that using attention weights directly as a measure of feature importance omits the indirect contribution of a feature in the attention mechanism. For the instance-wise first-order interpretation, we defined

$$FQ_i = \frac{\partial y}{\partial x_i} \tag{3.4}$$

as the full importance score for feature i,

$$DQ_i = k_i a_i \tag{3.5}$$

as the direct importance score for feature i, and

$$IQ_i = \sum_{j=1}^{d} \frac{\partial a_j}{\partial x_i} x_j \tag{3.6}$$

as the indirect importance score for feature i. For the model-wise first-order interpretation, we defined the model-wise full importance score (FP_i), direct importance score (DP_i), and indirect importance score (IP_i) for feature i as the averages of the absolute values of the corresponding instance-wise importance scores of this feature across all instances in the test set:

$$FP_i = |\overline{FQ_i}| \tag{3.7}$$

$$DP_i = |\overline{DQ_i}| \tag{3.8}$$

$$IP_i = |\overline{IQ_i}| \tag{3.9}$$

Because absolute values are used, the model-wise FP_i of feature i is no longer a sum of its IP_i and DP_i.

3.2.4 Second-order interpretation

It is computationally expensive and unscalable to compute the Hessian matrix for a large LINA model. Here, the Hessian matrix of the output w.r.t. the input feature vector is reducible to the Jacobian matrix of the attention vector w.r.t. the input feature vector in a LINA model, which is computationally feasible to calculate when the network utilizes leaky-ReLU or ReLU activation function. It is derived as follows:

$$\frac{\partial^2 y}{\partial x_i \partial x_j} = k_j \frac{\partial a_j}{\partial x_i} + k_i \frac{\partial a_i}{\partial x_j} \tag{3.10}$$

Proof:

$$\frac{\partial^2 y}{\partial x_i \partial x_j} = K^T \frac{\partial}{\partial x_i} \begin{bmatrix} x_1 \frac{\partial a_1}{\partial x_j} \\ \vdots \\ x_{j-1} \frac{\partial a_{j-1}}{\partial x_j} \\ x_j \frac{\partial a_j}{\partial x_j} + a_j \\ x_{j+1} \frac{\partial a_{j+1}}{\partial x_j} \\ \vdots \\ x_n \frac{\partial a_n}{\partial x_j} \end{bmatrix}$$

$$\frac{\partial^2 y}{\partial x_i \partial x_j} = K^T \begin{bmatrix} x_1 \frac{\partial^2 a_1}{\partial x_j \partial x_i} \\ \vdots \\ x_{j-1} \frac{\partial^2 a_{j-1}}{\partial x_j \partial x_i} \\ x_j \frac{\partial^2 a_j}{\partial x_j \partial x_i} + \frac{\partial a_j}{\partial x_i} \\ x_{j+1} \frac{\partial a_{j+1}}{\partial x_j \partial x_i} \\ \vdots \\ x_{i-1} \frac{\partial^2 a_{i-1}}{\partial x_j \partial x_i} \\ x_i \frac{\partial^2 a_i}{\partial x_j \partial x_i} + \frac{\partial a_i}{\partial x_j} \\ x_{i+1} \frac{\partial a_{i+1}}{\partial x_i \partial x_i} \\ \vdots \\ x_n \frac{\partial^2 a_n}{\partial x_j \partial x_i} \end{bmatrix}$$

We aim to demonstrate that, for any neuron, q, in the attention layer that outputs A (i.e., $q \in A$):

$$\frac{\partial^2 a_q}{\partial x_j \partial x_i} = 0$$

For any neuron $q \in A$:

$$a_q = \sum_{i=1}^{m_l} w_{q,k,l} f_{k,l}$$

$$\frac{\partial a_q}{\partial x_j} = \sum_{i=1}^{m_l} w_{q,k,l} \frac{\partial f_{k,l}}{\partial x_j}$$

$$\frac{\partial^2 a_q}{\partial x_i \partial x_j} = \sum_{i=1}^{m_l} w_{q,k,l} \frac{\partial^2 f_{k,l}}{\partial x_i \partial x_j}$$

Where $f_{k,l}$ is the activation function output from neuron k on hidden layer l containing m_l neurons, and $w_(i,k,l)$ the coefficient of the connection between neuron q on layer A and neuron k on layer l.

For this proof, we define the activation functions:

$ReLU(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases}$, and $Leaky-ReLU(x) = \begin{cases} x, & \text{if } f_{i,l} > 1 \\ -\alpha x, & \text{otherwise} \end{cases}$, where α is a constant.

Initial case:

Let's assume the case where MTL LINA has only one hidden layer. For any neuron q on the 1^{st} hidden layer, we have:

$$\frac{\partial f_{q,1}}{\partial x_j} = \frac{\partial f_{q,1}}{\partial o_{q,1}} \frac{\partial o_{q,1}}{\partial x_j}$$

.

with $o_{q,1}$ being the output of neuron q before activation.

$$o_{q,1} = \sum_{k=1}^{m} w_{q,k,1} x_k$$

Because $w_{q,k,1}$ is independent of x_j,

$$\frac{\partial o_{q,1}}{\partial x_j} = \sum_{k=1}^{m} w_{q,k,1} \frac{\partial x_k}{\partial x_j}$$

Then:

$$\frac{\partial f_{q,1}}{\partial x_j} = \frac{\partial f_{q,1}}{\partial o_{q,1}} \sum_{k=1}^{m} w_{q,k,l} \frac{\partial x_k}{\partial x_j}$$

$$\frac{\partial f_{q,1}}{\partial x_j} = \frac{\partial f_{q,1}}{\partial o_{q,1}} w_{q,j,1}$$

$$\frac{\partial^2 f_{q,1}}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i} (\frac{\partial f_{q,1}}{\partial o_{q,1}} w_{q,j,1})$$

$$\frac{\partial^2 f_{q,1}}{\partial x_i \partial x_j} = w_{q,j,1} \frac{\partial}{\partial x_i} (\frac{\partial f_{q,1}}{\partial o_{q,1}})$$

When $f_{q,1}$ is ReLU or leaky-ReLU, then $\frac{\partial}{\partial x_i}(\frac{\partial f_{q,1}}{\partial o_{q,1}}) = 0$ because for ReLU:$\frac{\partial f_{q,1}}{\partial o_{q,1}} = \begin{cases} 1, & \text{if } f_{q,1} > 0 \\ 0, & \text{else} \end{cases}$, or Leaky-ReLU: $\frac{\partial f_{q,1}}{\partial o_{q,1}} = \begin{cases} 1, & \text{if } f_{q,1} > 0 \\ -\alpha, & \text{else} \end{cases}$, and so the second-order derivative of those functions is assumed to be 0 everywhere. Thus:

$$\frac{\partial^2 f_{q,1}}{\partial x_i \partial x_j} = 0$$

And:

$$\frac{\partial^2 a_q}{\partial x_i \partial x_j} = \sum_{i=1}^{m_1} w_{q,k,1} \frac{\partial^2 f_{k,1}}{\partial x_i \partial x_j} = 0$$

Induction:

We hypothesize that, for a neural network with 2 or more hidden layers, we have at layer l, for any neuron q:

$$\frac{\partial^2 f_{q,l}}{\partial x_i \partial x_j} = 0$$

On the next hidden layer $l+1$, we have, for any neuron q:

$$\frac{\partial f_{q,l+1}}{\partial x_j} = \frac{\partial f_{q,l+1}}{\partial o_{q,l+1}} \frac{\partial o_{q,l+1}}{\partial x_j}$$

And:

$$o_{q,l+1} = \sum_{k=1}^{m_l} w_{q,k,l} f_{k,l}$$

Because $w_{q,k,l}$ is independent of x_j,

$$\frac{\partial o_{q,l+1}}{\partial x_j} = \sum_{k=1}^{m_l} w_{q,k,l} \frac{\partial f_{k,l}}{\partial x_j}$$

Then:

$$\frac{\partial f_{q,l+1}}{\partial x_j} = \frac{\partial f_{q,l+1}}{\partial o_{q,l+1}} \sum_{k=1}^{m} w_{q,k,l} \frac{\partial f_{k,l}}{\partial x_j}$$

$$\frac{\partial^2 f_{q,l+1}}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i} (\frac{\partial f_{q,l+1}}{\partial o_{q,l+1}} \sum_{k=1}^{m} w_{q,k,l} \frac{\partial f_{k,l}}{\partial x_j})$$

$$\frac{\partial^2 f_{q,l+1}}{\partial x_i \partial x_j} = \frac{\partial f_{q,l+1}}{\partial o_{q,l+1}} \sum_{k=1}^{m} w_{q,k,l} \frac{\partial}{\partial x_i} (\frac{\partial f_{k,l}}{\partial x_j}) + \frac{\partial}{\partial x_i} (\frac{\partial f_{q,l+1}}{\partial o_{q,l+1}}) \sum_{k=1}^{m} w_{q,k,l} \frac{\partial f_{k,l}}{\partial x_j}$$

$\frac{\partial}{\partial x_i} (\frac{\partial f_{q,l+1}}{\partial o_{q,l+1}}) = 0$ because the second derivative of ReLU or Leaky-ReLU is zero.

Thus,

$$\frac{\partial^2 f_{q,l+1}}{\partial x_i \partial x_j} = \frac{\partial f_{q,l+1}}{\partial o_{q,l+1}} \sum_{k=1}^{m} w_{q,k,l} \frac{\partial}{\partial x_i} (\frac{\partial f_{k,l}}{\partial x_j})$$

But for any neuron q on l (hypothesis):

$$\frac{\partial^2 f_{q,l}}{\partial x_i \partial x_j} = 0$$

By deduction:

$$\frac{\partial^2 f_{q,l+1}}{\partial x_i \partial x_j} = \frac{\partial f_{q,l+1}}{\partial o_{q,l+1}} \sum_{k=1}^{m} w_{q,k,l} 0$$

$$\frac{\partial^2 f_{q,l+1}}{\partial x_i \partial x_j} = 0$$

Conclusion:

By induction, we have demonstrated that for any neuron q on any layer l:

$$\frac{\partial^2 f_{q,l}}{\partial x_i \partial x_j} = 0$$

Therefore,

$$\frac{\partial^2 a_q}{\partial x_i \partial x_j} = \sum_{i=1}^{m_l} w_{q,k,l} \frac{\partial^2 f_{k,l}}{\partial x_i \partial x_j} = \sum_{i=1}^{m_l} w_{q,k,l} 0$$

$$\frac{\partial^2 a_q}{\partial x_i \partial x_j} = 0$$

For any $a_q \in A$:

$$\frac{\partial^2 y}{\partial x_i \partial x_j} = K^T \begin{bmatrix} 0 \\ \vdots \\ \frac{\partial a_j}{\partial x_i} \\ \vdots \\ \frac{\partial a_i}{\partial x_j} \\ \vdots \\ 0 \end{bmatrix}$$

Hence,

$$\frac{\partial^2 y}{\partial x_i \partial x_j} = k_j \frac{\partial a_j}{\partial x_i} + k_i \frac{\partial a_i}{\partial x_j}$$

End of Proof.

The K-weighted sum of the omitted second-order derivatives of the attention weights constitutes the approximation error. The performance of the second-order interpretation based on this approximation is benchmarked using synthetic and real-world datasets.

For instance-wise second-order interpretation, we define a directed importance score of feature r to feature c:

$$SQ_r^c = k_c \frac{\partial a_c}{\partial x_r} \tag{3.11}$$

This measures the importance of feature r in the calculation of the attention weight of feature c. In other words, this second-order importance score measures the importance of feature r to the direct importance score of feature c for the output. For the model-wise second-order interpretation, we defined an undirected importance score between feature r and feature c based on their average instance-wise second-order importance score in the test set:

$$SP_{c,r} = |\overline{SQ_r^c + SQ_c^r}| \tag{3.12}$$

3.2.5 Recap of the LINA importance scores

The notations and definitions of all the importance scores for a LINA model are recapitulated below. FQ and SQ are selected as the first-order and second-order importance scores, respectively, for instance-wise interpretation. FP and SP are used as the first-order and second-order importance scores, respectively, for model-wise interpretation.

Order	Target	Acronym	Definition
First-order	Instance-wise	FQ	$\mathrm{FQ_i} = \mathrm{DQ_i} + \mathrm{IQ_i}$
		DQ	$\mathrm{DQ_i} = k_i a_i$
		IQ	$\mathrm{IQ_i} = \sum_{c=1}^{d} \mathrm{SQ}_i^c \, x_c$
	Model-wise	FP	$\mathrm{FP_i} = \overline{\lvert \mathrm{FQ_i} \rvert}$
		DP	$\mathrm{DP_i} = \overline{\lvert \mathrm{DQ_i} \rvert}$
		IP	$\mathrm{IP_i} = \overline{\lvert \mathrm{IQ_i} \rvert}$
Second-order	Instance-wise	SQ	$\mathrm{SQ_r^c} = k_c \frac{\partial a_c}{\partial x_r}$
	Model-wise	SP	$\mathrm{SP_{c,r}} = \overline{\lvert SQ_r^c + SQ_c^r \rvert}$

3.3 Data and Experimental Setup

3.3.1 California housing dataset

The California housing dataset (Pace and Barry, 1997) was used to formulate a simple regression task, which is the prediction of the median sale price of houses in a district based on eight input features (Table 3.1). The dataset contained 20640 instances (districts) for model training and testing.

3.3.2 First-order benchmarking datasets

Five synthetic datasets, each containing 20,000 instances, were created using the sigmoid functions to simulate binary classification tasks. These functions were created

Features \ Outputs	Linearization Output				First-order Instance-wise Importance Scores		
	Coefficients (K)	Attention (A)	Features (X)	Products (KAX)	FQ	DQ	IQ
longitude	249	221	0.22	11,932	-51,296	55,108	-106,404
latitude	257	-299	-0.56	42,700	-211,275	-76,933	-134,343
median_age	213	-275	-1.35	79,230	33,407	-58,524	91,930
total_rooms	173	158	1.32	36,024	-17,957	27,230	-45,187
total_bedrooms	184	240	1.10	48,531	5,614	44,281	-38,667
population	200	-19	1.54	-5,690	-62,220	-3,695	-58,525
households	189	233	1.20	52,532	32,443	43,951	-11,508
median_income	174	125	0.91	19,777	73,337	21,736	51,601
bias				149			
median_house_price				285,183			

Table 3.1: **The linearization outputs and first-order instance-wise importance scores for a district from the California housing dataset.**

following the examples in (Chen et al., 2018) for the first-order interpretation benchmarking. All five datasets included ten input features. The values of the input features were independently sampled from a standard Gaussian distribution: $x_i \sim N(0,1), i \in \{1,2,\ldots,10\}$. The target value was set to 0 if the sigmoid function output is (0,0.5). The target value was set to 1, if the sigmoid function output is [0.5,1). We used the following five sigmoid functions of different subsets of the input features:

(F1): $Sig(4X_1^2 - 3X_2^2 - 2X_3^2 + X_4^2)$. This function contains four important features with independent squared relationships with the target. The ground-truth rankings of the features by first-order importance are X_1, X_2, X_3, and X_4. The remaining six uninformative features are tied in the last rank.

(F2): $Sig(-10\sin(X_1) + 2|X_2| + X_3 - \exp(-X_4))$. This function contains four important features with various non-linear additive relationships with the target. The ground-truth ranking of the features is X_1, X_4, X_2, and X_3. The remaining six uninformative features are tied in the last rank.

(F3): $Sig(4X_1X_2X_3 + X_4X_5X_6)$. This function contains six important features with multiplicative interactions among one another. The ground-truth ranking of the features is X_1, X_2, and X_3 tied in the first rank, X_4, X_5, and X_6 tied in the second rank, and the remaining uninformative features tied in the third rank.

(F4): $Sig(-10\sin(X_1X_2X_3) + |X_4X_5X_6|)$. This function contains six important features with multiplicative interactions among one another and non-linear relationships with the target. The ground-truth ranking of the features is X_1, X_2, and X_3 tied in the first rank, X_4, X_5, and X_6 tied in the second rank, and the other four uninformative features tied in the third rank.

(F5): $Sig(-20sin(X_1X_2) + 2|X_3| + X_4X_5 - 4exp(-X_6))$. This function contains six important features with a variety of non-linear relationships with the target. The

ground-truth ranking of the features is X_1 and X_2 tied in the first rank, X_6 in the second, X_3 in the third, X_4 and X_5 tied in the fourth, and the remaining uninformative features tied in the fifth.

3.3.3 Second-order benchmarking dataset

Ten regression synthetic datasets, referred to as F6-A, F7-A, F8-A, F9-A, and F10-A (-A datasets) and F6-B, F7-B, F8-B, F9-B, and F10-B (-B datasets) were created. The -A datasets followed the examples in (Tsang et al., 2017) for the second-order interpretation benchmarking. The -B datasets used the same functions below to compute the target as the -A datasets, but included more uninformative features to benchmark the interpretation performance on high-dimensional data. Each -A dataset contained 5,000 instances. Each -B dataset contained 10,000 instances. The five -A datasets included 13 input features. The five -B datasets included 100 input features, some of which were used to compute the target. In F7-A/B, F8-A/B, F9-A/B, and F10-A/B, the values of the input features of an instance were independently sampled from a standard uniform distribution: $X_i \sim U(-1, 1)$,$i \in \{1, 2, \ldots, 13\}$ in the -A datasets or $i \in \{1, 2, \ldots, 100\}$ in the -B datasets. In the F6 dataset, the values of the input features of an instance were independently sampled from two uniform distributions: $X_i \sim U(0, 1)$, $i \in 1, 2, 3, 6, 7, 9, 11, 12, 13\}$ in the -A datasets and $i \in \{1, 2, 3, 6, 7, 9, 11, ..., 100\}$ in the -B datasets; and $X_i \sim U(0.6, 1)$, $i \in \{4, 5, 8, 10\}$ in both. The value of the target for an instance was computed using the following five functions:

(F6-A) and (F6-B): $\pi^{X_1 X_2}\sqrt{X_3} + \sin^{-1} X_4 + \log(X_3 + X_5) + \frac{X_9}{X_{10}}\sqrt{\frac{X_7}{X_8}} - X_2 X_7$. This function contains eleven pairwise feature interactions: $\{(X_1, X_2), (X_1, X_3), (X_2, X_3), (X_3, X_5), (X_7, X_8), (X_7, X_9), (X_7, X_10), (X_8, X_9), (X_8, X_{10}), (X_9, X_{10}), (X_2, X_7)\}$.

(F7-A) and (F7-B): $\exp(|X_1 - X_2|) + |X_2 X_3| - X_3^{2|X_4|} + \log(X_4^2 + X_5^2 + X_7^2 + X_8^2) + X_9 +$

$\frac{X_{10}^2}{1+X_{10}^2}$. This function contains nine pairwise interactions: $\{(X_1, X_2), (X_2, X_3), (X_3, X_4), (X_4, X_5), (X_4, X_7), (X_4, X_8), (X_5, X_7), (X_5, X_8), (X_7, X_8)\}$.

(F8-A) and (F8-B): $\sin(|X_1 X_2| + 1) - \log(|X_3 X_4| + 1) + \cos(X_5 + X_6 - X_8) + \sqrt{X_8^2 + X_9^2 + X_{10}^2}$. This function contains ten pairwise interactions: $\{(X_1, X_2), (X_3, X_4), (X_5, X_6), (X_4, X_7), (X_5, X_6), (X_5, X_8), (X_6, X_8), (X_8, X_9), (X_8, X_{10}), (X_9, X_{10})\}$.

(F9-A) and (F9-B): $\tanh(X_1 X_2 + X_3 X_4)\sqrt{|X_5|} + \log\left[(X_6 X_7 X_8)^2 + 1\right] + X_9 X_{10} + \frac{1}{1+|X_{10}|}$. This function contains thirteen pairwise interactions: $\{(X_1, X_2), (X_1, X_3), (X_2, X_3), (X_2, X_4), (X_3, X_4), (X_1, X_5), (X_2, X_5), (X_3, X_5), (X_4, X_5), (X_6, X_7), (X_6, X_8), (X_7, X_8), (X_9, X_{10})\}$.

(F10-A) and (F10-B): $\cos(X_1 X_2 X_3) + \sin(X_4 X_5 X_6)$. This function contains six pairwise interactions: $\{(X_1, X_2), (X_1, X_3), (X_2, X_3), (X_4, X_5), (X_4, X_6), (X_5, X_6)\}$.

3.3.4 Breast cancer dataset

The Discovery, Biology, and Risk of Inherited Variants in Breast Cancer (DRIVE) project (Amos et al., 2017) generated a breast cancer dataset (NIH dbGaP accession number: phs001265.v1.p1) for genome-wide association study (GWAS) and predictive genomics. This cohort contained 26,053 case subjects with malignant tumor or in situ tumor and 23,058 control subjects with no tumor. The task for predictive genomics is a binary classification of subjects between cases and controls. The breast cancer dataset was processed using PLINK (Purcell et al., 2007) as described previously (Badré et al., 2021) to compute the statistical significance of the SNPs. Out of a total of 528,620 SNPs, 1541 SNPs had a p-value lower than 10^{-6} and were used as the input features for predictive genomics. To benchmark the performance of the model interpretation, 1541 decoy SNPs were added as input features. The frequencies of homozygous minor alleles, heterozygous alleles, and homozygous dominant alleles were the same between decoy SNPs and real SNPs. Because decoy SNPs have random relationships with

the case/control phenotype, they should not be selected as important features or be included in salient interactions by model interpretation.

3.3.5 Implementations and Evaluation Strategies

The California Housing Dataset was partitioned into a training set (70%), a validation set (20%), and a test set (10%). The eight input features were longitude, latitude, median age, total rooms, total bedrooms, population, households, and median income. The median house value was the target of the regression. All the input features were standardized to zero mean and unit standard deviation based on the training set. Feature standardization is critical for model interpretation in this case because the scale for the importance scores of a feature is determined by the scale for the values of this feature, and comparison of the importance scores between features requires the values of the features to be in the same scale. The LINA model comprised an input layer (8 neurons), five fully connected hidden layers (7, 6, 5, 4, and 3 neurons), and an attention layer (8 neurons) for the inner attention neural network, followed by a second input layer (8 neurons), a linearization layer (8 neurons), and an output layer (1 neuron). The hidden layers used ReLU as the activation function. No regularization was applied to the coefficient vector and L1 regularization was applied to the attention vector ($\gamma = 10^{-6}$). The LINA model was trained using the Adam optimizer with a learning rate of 10^{-2}. The predictive performance of the obtained LINA model was benchmarked to have an RMSE of 71055 in the test set. As a baseline model for comparison, a gradient boosting model achieved an RMSE of 77852 in the test set using 300 decision trees with a maximum depth of 5.

For the first-order interpretation, each synthetic dataset was split into a cross-validation set (80%) for model training and hyperparameter optimization and a test set (20%) for

performance benchmarking and model interpretation. A LINA model and a feedforward neural network (FNN) model were constructed using 10-fold cross-validation. For the first four synthetic datasets, the inner attention neural network in the LINA model had 3 layers containing 9 neurons in the first layer, 5 neurons in the second layer, and 10 neurons in the attention layer. The FNN had 3 hidden layers with the same number of neurons in each layer as the inner attention neural network in the LINA model. For the fifth function with more complex relationships, the first and second layers were widened to 100 and 25 neurons, respectively, in both the FNN and LINA models to achieve a predictive performance similar to the other datasets in their respective validation sets. Both the FNN and LINA models were trained using the Adam optimizer. The learning rate was set to 10^{-2}. The mini-batch size was set to 32. No hyperparameter tuning was performed. The LINA model was trained with the L2 regularization on the coefficient vector ($\beta = 10^{-4}$) and the L1 regularization on the attention vector ($\gamma = 10^{-6}$). The values of β and γ were selected from 10^{-2},10^{-3},10^{-4},10^{-5},10^{-6},10^{-7}, and 0 based on the predictive performance of the LINA model on the validation set. Batch normalization was used for both architectures. Both the FNN and LINA models achieved predictive performance at approximately 99% AUC on the test set in the five first-order synthetic datasets, which was comparable to (Chen et al., 2018). Deep Lift (Shrikumar et al., 2017), LIME (Ribeiro et al., 2016), Grad*Input (Shrikumar et al., 2017), L2X (Chen et al., 2018) and Saliency (Simonyan et al., 2014) were used to interpret the FNN model and calculate the feature importance scores using their default configurations. FP, DP, and IP scores were used as the first-order importance scores for the LINA model. We compared the performances of the first-order interpretation of LINA with DeepLIFT, LIME, GradInput, and L2X. The interpretation accuracy was measured using the Spearman rank correlation coefficient between the predicted ranking of features by their first-order importance and the ground-truth ranking. This

metric was chosen because it encompasses both the selection and ranking of the important features.

For the second-order interpretation benchmarking, each synthetic dataset was also split into a cross-validation set (80%) and a test set (20%). A LINA model, an FNN model for NID, and a Bayesian neural network (BNN) for GEH, as shown in (Cui et al., 2019), were constructed based on the neural network architecture used in (Tsang et al., 2017) using 10-fold cross-validation. The inner attention neural network in the LINA model uses 140 neurons in the first hidden layer, 100 neurons in the second hidden layer, 60 neurons in the third hidden layer, 20 neurons in the fourth hidden layer, and 13 neurons in the attention layer. The FNN model was composed of 4 hidden layers with the same number of neurons in each layer as LINA's inner attention neural network. The BNN model uses the same architecture as that of the FNN model. The FNN, BNN, and LINA models were trained using the Adam optimizer with a learning rate of 10^{-3} and a mini-batch size of 32 for the -A datasets and 128 for the -B datasets. The LINA model was trained using L2 regularization on the coefficient vector ($\beta = 10^{-4}$) and the L1 regularization on the attention vector ($\gamma = 10^{-6}$) with batch normalization. Hyperparameter tuning was performed as described above to optimize the predictive performance. The FNN and BNN models were trained using the default regularization parameters, as shown in (Cui et al., 2019), (Tsang et al., 2017). Batch normalization was used for LINA. The FNN, BNN, and LINA models all achieved R^2 scores of more than 0.99 on the test sets of the five -A datasets, as in the examples in (Tsang et al., 2017), while their R^2 scores ranged from 0.91 to 0.93 on the test set of the five high-dimensional -B datasets. Pairwise interactions in each dataset were identified from the BNN model using GEH (Cui et al., 2019), the FNN model using NID (Tsang et al., 2017), and the LINA model using the SP scores. For GEH, the number of clusters was set to the number of features and the number of iterations was set to 20. NID

was run using its default configuration. For a dataset with m pairs of ground-truth interactions, the top-m pairs with the highest interaction scores were selected from each algorithm's interpretation output. The percentage of ground-truth interactions in the top-m predicted interactions (i.e., the precision) was used to benchmark the second-order interpretation performance of the algorithms.

For the breast cancer dataset, 49111 subjects in the breast cancer dataset were randomly divided into the training set (80%), validation set (10%), and test set (10%). The FNN model and the BNN model had 3 hidden layers with 1000, 250, and 50 neurons as described previously (Badré et al., 2021). The same hyperparameters were used in our previous study (Badré et al., 2021). The inner attention neural network in the LINA model also used 1000, 250, and 50 neurons before the attention layer. All of these models had 3082 input neurons for 1541 real SNPs and 1541 decoy SNPs. β was set to 0.01 and γ to 0, which were selected from 10^{-2},10^{-3},10^{-4},10^{-5},10^{-6},10^{-7}, and 0 based on the predictive performance of the LINA model on the validation set. Early stopping based on the validation AUC score was used during training. The FNN, BNN, and LINA models achieved a test AUC of 64.8%, 64.8%, and 64.7% on the test set, respectively, using both the 1541 real SNPs with p-values less than 10^{-6} and the 1541 decoy SNPs. The test AUCs of these models were lower than that of the FNN model in our previous study (Badré et al., 2021) at 67.4% using real 5,273 SNPs with p-values less than 10^{-3} as input. As the same FNN architecture design was used in the two studies, the reduction in the predictive performance in this study can be attributed to the use of more stringent p-value filtering to retain only real SNPs with a high likelihood of having a true association with the disease and the addition of decoy SNPs for benchmarking the interpretation performance.

Deep Lift (Shrikumar et al., 2017), LIME (Ribeiro et al., 2016), Grad*Input (Shrikumar et al., 2017), L2X (Chen et al., 2018) and Saliency (Simonyan et al., 2014) were

used to interpret the FNN model and calculate the feature importance scores using their default configurations. The FP score was used as the first-order importance score for the LINA model. After the SNPs were filtered at a given importance score threshold, the false discovery rate (FDR) was computed from the retained real and decoy SNPs above the threshold. The number of retained real SNPs was the total positive count for the FDR. The number of false positive hits (i.e., the number of unimportant real SNPs) within the retained real SNPs was estimated as the number of retained decoy SNPs. Thus, FDR was estimated by dividing the number of retained decoy SNPs by the number of retained real SNPs. An importance-score-sorted list of SNPs from each algorithm was filtered at an increasingly stringent score threshold until reaching the desired FDR level. The interpretation performance of an algorithm was measured by the number of top-ranked features filtered at 0.1%, 1%, and 5% FDR and the FDRs for the top-100 and top-200 SNPs ranked by an algorithm. For the second-order interpretation, pairwise interactions were identified from the BNN model using GEH (Cui et al., 2019), from the FNN model using NID (Tsang et al., 2017), and from the LINA model using the SP scores. For GEH, the number of clusters was set to 20 and the number of iterations was set to 20. While LINA and NID used all 4,911 subjects in the test set and completed their computation within an hour, the GEH results were computed for only 1000 random subjects in the test set over ¿2 days because GEH would have taken approximately two months to complete the entire test set with its n2 computing cost where n is the number of subjects. NID was run using its default configuration in the FNN model. The interpretation accuracy was also measured by the numbers of top-ranked pairwise interactions detected at 0.1%, 1%, and 5% FDR and the FDRs for the top-1000 and top-2000 interaction pairs ranked by an algorithm. A SNP pair was considered to be false positive if one or both of the SNPs in a pair was a decoy.

3.4 Results and Discussion

3.4.1 Demonstration of LINA on a real-world application

In this section, we demonstrate LINA using the California housing dataset, which has been used in previous model interpretation studies for algorithm demonstration (Cui et al., 2019), (Tsang et al., 2017). Four types of interpretations from LINA were presented, including the instance-wise first-order interpretation, the instance-wise second-order interpretation, the model-wise first-order interpretation, and the model-wise second-order interpretation.

3.4.1.1 Instance-wise Interpretation

Table 3.1 shows the prediction and interpretation results of the LINA model for an instance (district # 20444) that had a true median price of $208600. The predicted price of $285183 was simply the sum of the eight element-wise products of the attention, coefficient, and feature columns plus the bias. This provided an easily understandable representation of the intermediate computation behind the prediction for this instance. For example, the median age feature had a coefficient of 213 in the model. For this instance, the median age feature had an attention weight of -275, which switched the median age to a negative feature and amplified its direct effect on the predicted price in this district. The product of the attention weight and coefficient yielded the direct importance score of the median age feature (i.e., DQ = -58,524), which represented the strength of the local linear association between the median age feature and the predicted price for this instance. By assuming that the attention weights of this instance are fixed, one can expect a decrease of $58,524 in the predicted price for an increase in the median age by one standard deviation (12.28 years) for this district. But this did not consider the effects of the median age increase on the attention weights,

which was accounted for by its indirect importance score (i.e., IQ = 91,930). The positive IQ indicated that a higher median age would increase the attention weights of other positive features and increase the predicted price indirectly. Combining the DQ and IQ, the positive FQ of 33,407 marked the median age to be a significant positive feature for the predicted price, perhaps through the correlation with some desirable variables for this district. This example suggested a limitation of using the attention weights themselves to evaluate the importance of features in the attentional architectures. The full importance scores represented the total effect of a feature's change on the predicted price. For this instance, the latitude feature had the largest impact on the predicted price. Table 3.2 presents a second-order interpretation of the prediction for this instance. The median age row in Table 3.2 shows how the median age feature impacted the attention weights of the other features. The two large positive SQ values of median age to the latitude and longitude features indicated significant increases of the two location features' attention weights with the increase of the median age. In other words, the location becomes a more important determinant of the predicted price for districts with older houses. The total bedroom feature received a large positive attention weight for this instance. The total bedroom column in Table 3.2 shows that the longitude and latitude features are the two most important determinants for the attention weights of the total bedroom feature. This suggested how a location change may alter the direct importance of the total bedroom feature for the price prediction of this district.

3.4.1.2 Model-wise Interpretation

Figure 3.2 shows the first-order model-wise interpretation results across districts in the California Housing dataset. The longitude, latitude, and population were the three

Column features (c) / Row features (r)	longitude	latitude	median_ age	total_ rooms	total_ bedrooms	population	households	median_ income
longitude	-17,234	-33,983	19,682	-10,797	-9,572	-13,375	-1,153	4,899
latitude	22,696	44,572	-25,631	13,068	12,002	18,119	1,035	-10,005
median_age	18,591	18,555	-14,262	7,140	5,749	8,328	2,586	-8,357
total_rooms	-13,249	-17,930	11,547	-4,102	-4,198	-8,626	-526	12,029
total_bedrooms	-16,973	-19,799	14,110	-7,173	-5,943	-8,597	-2,123	7,328
population	932	11,223	-4,307	1,052	1,947	4,842	-1,471	-4,623
households	-18,646	-25,227	16,879	-8,943	-7,507	-10,514	-2,163	6,829
median_income	-8,713	-20,704	10,829	-3,928	-4,515	-9,182	758	9,546

Table 3.2: **Second-order instance-wise importance scores of feature r (row r) to feature c (column c):** $SQ_r^c = k_c \frac{\partial a_c}{\partial x_r}$

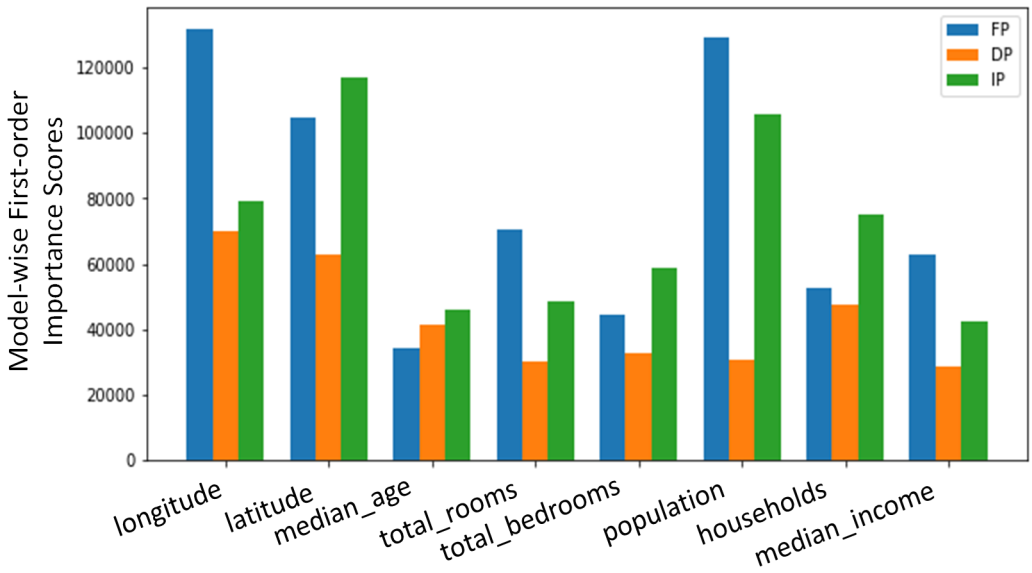


Figure 3.2: **First-order model-wise interpretation.** The three bars of a feature represented the FP, IP, and DP scores of this feature in the LINA model.

most important features. The longitude and latitude had both high direct importance scores and high indirect importance scores. However, the population feature derived its importance mostly from its heavy influence on the attention weights as measured by its indirect importance score. Figure 3.3 shows the second-order model-wise interpretation results for pairs of different features. Among all the feature pairs, the latitude and longitude features had the most prominent interactions, which was reasonable because the location was jointly determined by these two features. Some significant differences existed between the instance-wise interpretation and model-wise interpretation (e.g., Table 3.1 vs. Figure 3.2 and Table 3.2 vs. Figure 3.3). This illustrates the need for both instance-wise and model-wise interpretation methods for different purposes

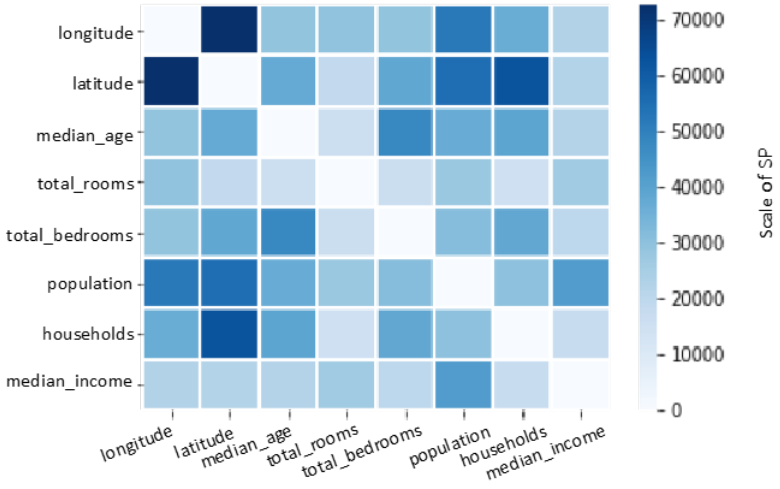


Figure 3.3: **Second-order model-wise interpretation.** The second-order model-wise importance scores (SP) are undirected between two features and are shown in a symmetric matrix as a heatmap. The importance scores for the feature self-interactions are set to zero in the diagonal of the matrix.

3.4.2 Benchmarking of the first-order and second-order interpretation using synthetic datasets

In real-world applications, the true importance of features for prediction cannot be determined with certainty and may vary among different models. Therefore, previous studies on model interpretation (Ribeiro et al., 2016), (Cui et al., 2019) benchmarked their interpretation performance using synthetic datasets with known ground-truth of feature importance. In this study, we also compared the interpretation performance of LINA with the SOTA methods using synthetic datasets created as in previous studies (Chen et al., 2018), (Tsang et al., 2017).

The performance of the first-order interpretation of LINA was compared with DeepLIFT, LIME, Grad*Input, and L2X (Table 3.3). The three first-order importance scores from LINA, including FP, DP, and IP, were tested. The DP score performed the

Datasets / Methods	F1	F2	F3	F4	F5	Average
LINA DP	**1.00±0.00**	0.88±0.03	0.25±0.07	0.65±0.05	0.92±0.03	0.74±0.04
LINA IP	**1.00±0.00**	0.92±0.03	0.69±0.01	0.84±0.03	0.96±0.03	0.88±0.02
LINA FP	**1.00±0.00**	0.97±0.02	**1.00±0.00**	**0.91±0.04**	**1.00±0.00**	**0.98±0.01**
DeepLift	0.99±0.01	**1.00±0.00**	0.95±0.03	0.83±0.12	**1.00±0.00**	0.95±0.03
Saliency	**1.00±0.00**	0.90±0.01	**1.00±0.00**	0.76±0.11	**1.00±0.00**	0.93±0.03
Grad*Input	**1.00±0.00**	**1.00±0.00**	0.85±0.08	0.78±0.12	**1.00±0.00**	0.93±0.04
L2X	0.59±0.06	0.41±0.07	0.15±0.11	0.30±0.08	0.5±0.03	0.39±0.07
LIME	-0.72±0.0	-0.52±0.08	-0.14±0.07	-0.57±0.05	-0.3±0.06	-0.45±0.05

*The best Spearman correlation coefficient for each synthetic dataset is highlighted in bold

Table 3.3: **Benchmarking of the first-order interpretation performance using five synthetic datasets (F1 to F5)***

worst among the three, especially in the F3 and F4 datasets which contained interactions among three features. This suggested the limitation of using attention weights as a measure of feature importance. The FP score provided the most accurate ranking among the three LINA scores because it accounted for the direct contribution of a feature and its indirect contribution through attention weights. The first-order importance scores were then compared among different algorithms. L2X and LIME distinguished many important features correctly from un-informative features, but their rankings of the important features were often inaccurate. The gradient-based methods produced mostly accurate rankings of the features based on their first-order importance. Their interpretation accuracy generally decreased in datasets containing interactions among more features. Among all the methods, the LINA FP scores provided the most accurate ranking of the features on average.

The performance of the second-order interpretation of LINA was compared with those of GEH and NID (Table 3.4). There were a total of 78 possible pairs of interactions among 13 features in each -A synthetic dataset and there were 4950 possible pairs of interactions among 100 features in each -B synthetic dataset. The precision from random guesses was only $\sim 12.8\%$ on average in the -A datasets and less than 1% in the -B datasets. The three second-order algorithms all performed significantly better than the random guess. In the -A datasets, the average precision of LINA SP was $\sim 80\%$, which was $\sim 12\%$ higher than that of NID and $\sim 29\%$ higher than that of GEH. The addition of 87 un-informative features in the -B datasets reduced the average precision of LINA by $\sim 15\%$, that of NID by $\sim 13\%$, and that of GEH by $\sim 22\%$. In the -B datasets, the average precision of LINA SP was $\sim 65\%$, which was $\sim 9\%$ higher than that of NID and $\sim 35\%$ higher than that of GEH. This indicates that more accurate second-order interpretations can be obtained from the LINA models.

Total Features	Datasets	NID	GEH	LINA SP
13 features	F6-A	44.5%±0.2%	50.0%±0.2%	**61.8%±0.2%**
	F7-A	**98.0%±0.1%**	41.0%±0.2%	92.0%±0.1%
	F8-A	80.6%±0.2%	48.8%±0.4%	**85.0%±0.2%**
	F9-A	62.2%±0.4%	41.4%±0.3%	**70.0±0.3%**
	F10-A	56.7%±0.3%	75.0%±0.5%	**91.7%±0.3%**
	Average	68.4%±0.2%	51.2%±0.3%	**80.1±0.2%**
100 features	F6-B	51.8%±0.2%	18.1%±1.0%	**52.7%±0.3%**
	F7-B	44.0%±0.2%	28.8%±0.4%	**90.0%±0.0%**
	F8-B	76.3%±0.1%	47.9%±0.2%	**80%.0±0.3%**
	F9-B	40.0%±0.3%	41.8%±0.2%	**51.7%±0.3%**
	F10-B	**66.6%±0.0%**	10.4%±1.0%	50.0%±0.1%
	Average	55.7%±0.2%	29.4%±0.6%	**64.9%±0.2%**

*The best precision for each dataset is highlighted in bold

Table 3.4: **Precision of the second-order interpretation by LINA SP, NID and GEH in ten synthetic datasets (F6 to F10)***

3.4.3 Benchmarking of the first-order and second-order interpretation using a predictive genomics application

As the performance benchmarks in synthetic datasets may not reflect those in real-world applications, we engineered a real-world benchmark based on a breast cancer dataset for predictive genomics. While it was unknown which SNPs and which SNP interactions were truly important for phenotype prediction, the decoy SNPs added by us were truly unimportant. Moreover, a decoy SNP cannot have a true interaction, such as XOR or multiplication, with a real SNP to have a joint impact on the disease outcome. Thus, if a decoy SNP or an interaction with a decoy SNP is ranked by an algorithm as important, it should be considered a false positive detection. As the number of decoy SNPs was the same as the number of real SNPs, the false discovery rate can be estimated by assuming that an algorithm makes as many false positive detections from the decoy SNPs as from the real SNPs. This allowed us to compare the number of positive detections by an algorithm at certain FDR levels.

The first-order interpretation performance of LINA was compared with those of DeepLIFT, LIME, Grad*Input, and L2X (Table 3.5). At 0.1%, 1%, and 5% FDR, LINA identified more important SNPs than other algorithms. LINA also had the lowest FDRs for the top-100 and top-200 SNPs. The second-order interpretation performance of LINA was compared with those of NID and GEH (Table 3.6). At 0.1%, 1%, and 5% FDR, LINA identified more pairs of important SNP interactions than NID and GEH did. LINA had lower FDRs than the other algorithms for the top-1000 and top-2000 SNP pairs. Both L2X and GEH failed to output meaningful importance scores in this predictive genomics dataset. Because GEH needed to compute the full Hessian, it was also much more computationally expensive than the other algorithms.

Methods	LINA FP	Saliency	grad*Input	DeepLift	LIME	L2X
# SNPs at 0.1% FDR	**127**	35	75	75	9	0
# SNPs at 1% FDR	**158**	35	88	85	9	0
# SNPs at 5% FDR	**255**	57	122	119	9	0
FDR at top-100 SNP	**0.0%**	7.5%	3.0%	2.0%	16.3%	N/A
FDR at top-200 SNP	**1.5%**	16.2%	9.3%	9.3%	20.5%	N/A

Table 3.5: **Performance benchmarking of the first-order interpretation for predictive genomics**

Methods	LINA SP	NID	GEH
# SNP pairs at 0.1% FDR	**583**	415	0
# SNP pairs at 1% FDR	**1040**	504	0
# SNP pairs at 5% FDR	**2887**	810	0
FDR at top-1000 SNP pairs	**0.9%**	10.5%	N/A
FDR at top-2000 SNP pairs	**3.0%**	31.8%	N/A

Table 3.6: **Performance benchmarking of the second-order interpretation for predictive genomics**

The existing model interpretation algorithms and LINA can provide rankings of the features or feature interactions based on their importance scores at arbitrary scales. We demonstrated that decoy features can be used in real-world applications to set thresholds for first-order and second-order importance scores based on the FDRs of retained features and feature pairs. This provided an uncertainty quantification of the model interpretation results without knowing the ground-truth in real-world applications. The predictive genomics application provided a real-world test of the interpretation performance of these algorithms. In comparison with the synthetic datasets, the predictive genomics dataset was more challenging for model interpretation, because of the low predictive performance of the models and the large number of input features. For this real-world application, LINA was shown to provide better first-order and second-order interpretation performance than existing algorithms on a model-wise level. Furthermore, LINA can provide instance-wise interpretation to identify important SNP and SNP interactions for the prediction of individual subjects. Model interpretation is important for making biological discoveries from predictive models, because first-order interpretation can identify individual genes involved in a disease ((Rivandi et al., 2018; Romualdo Cardoso et al., 2022)), and second-order interpretation can uncover epistatic interactions among genes for a disease ((Shaker and Senousy, 2019; van de Haar et al., 2019)). These discoveries may provide new drug targets ((Wang et al., 2018; Gao et al., 2019; Gonçalves et al., 2020)) and enable personalized formulation of treatment plans ((Wu et al., 2016; Zhao et al., 2021; Velasco-Ruiz et al., 2021)) for breast cancer.

3.5 Conclusion

In this study, we designed a new neural network architecture, referred to as LINA, for model interpretation. LINA uses a linearization layer on top of a deep inner attention

neural network to generate a linear representation of model prediction. LINA provides the unique capability of offering both first-order and second-order interpretations and both instance-wise and model-wise interpretations. The interpretation performance of LINA was benchmarked to be higher than the existing algorithms on synthetic datasets and a predictive genomics dataset.

Chapter 4

Explainable multi-task learning improves the parallel estimation of polygenic risk scores for many diseases through shared genetic basis

4.1 Introduction

The polygenic risk score (PRS) of a complex disease quantifies the genetic risk of an individual for this disease based on many genetic variants across the whole genome of this individual. The risk variants are generally selected based on this disease's genome-wide association studies (GWAS), often using a relaxed statistical significance threshold. A PRS can be estimated using a variety of statistical methods, including Best Linear Unbiased Prediction (BLUP) (Whittaker et al., 2000b; Meuwissen et al., 2001; Clark et al., 2013; Maier et al., 2015; Speed and Balding, 2014) and LDPred (Privé et al., 2020; Vilhjálmsson et al., 2015; Khera et al., 2018; Márquez-Luna et al., 2021). Statistical models of PRS have been built for breast cancer (Khera et al., 2018), colorectal cancer[(Thomas et al., 2020), (Gola et al., 2020), Type-2 diabetes (Ge et al., 2022), cardiovascular disease (Ye et al., 2021), and many other diseases. These statistical methods generally assume that the effects of risk variants on a phenotype are linear and independent. Recently, machine learning approaches free of these assumptions (Ho et al., 2019) have been used to estimate the PRS for breast cancer (Badré et al., 2021), blood pressure (Elgart et al., 2022), and schizophrenia (Bracher-Smith et al., 2022). However, the existing studies generally focused on constructing independent PRS models for individual diseases.

Many complex diseases share a substantial amount of common risk genetic determinants. Genome-wide cross-trait analyses have been performed between obesity and cardiovascular diseases (Zhuang et al., 2021), between thyroid and breast cancers (Sutton et al., 2022), between uterine leiomyoma and breast cancer (Wu et al., 2022), between asthma and cardiovascular diseases (Zhou et al., 2022), between Alzheimer's disease and gastrointestinal tract disorders (Adewuyi et al., 2022), between Alzheimer disease and major depressive disorder (Lutz et al., 2020), between lung cancer and

chronic bronchitis (Byun et al., 2021), and so on. These studies were often motivated by frequent co-occurrences of a pair of diseases in a population. Some of the epidemiological associations have been attributed to the shared genetic architecture between the diseases. The related genetic etiology among diseases can be caused by dysfunctions in some common enzymes or pathways, which may increase the clinical risks for multiple diseases directly or indirectly.

In this study, we hypothesized that shared genetic determinants among diseases can be exploited to improve their PRS estimation. We tested this hypothesis using a pan-disease multi-task learning (MTL) approach (Caruana, 1998) based on an interpretable neural network architecture (Badré and Pan, 2022). MTL has been widely used in many computer vision (Girshick, 2015) and natural text processing (Liu et al., 2016) applications, in which the training examples have multiple labels to be predicted from the same input feature vectors. Unlike single-task learning (STL), which trains a model to predict each individual label independently, MTL trains a model to predict all labels in parallel. MTL has been shown to provide better predictive performance than STL when the learning tasks are related (Standley et al., 2019). Related tasks can enable a MTL model to learn a better-shared representation through data amplification, feature selection, regularization, and other beneficial effects (Fifty et al., 2021). However, if the tasks are unrelated, the predictive performance of MTL may be worse than that of STL, owing to the negative knowledge transfer among the tasks (Standley et al., 2019). Thus, if our hypothesis is invalid, the PRS learned for a disease in conjunction with other diseases by a pan-disease MTL model would be less accurate than the PRS learned for this disease by an STL model.

4.2 Methods

4.2.1 Preparation of the phenotypic and genomic data

A total of 488,175 subjects were extracted from the UK Biobank dataset release version 2 (Bycroft et al., 2018). The phenotypic traits of the subjects were determined using the protocol and software described in a previous study (DeBoever et al., 2020). The diseases in subjects were identified using hospital inpatient records (ICD10 codes, UK Biobank Data Coding 19) and self-reported disease status (UK Biobank Data Coding 3 for cancers and UK Biobank Data Coding 6 for non-cancer diseases). The UKB genomic data covered a total of 805,426 SNPs. The genotypes of SNPs were encoded as 0 for homozygous with the minor allele, 1 for heterozygous alleles, or 2 for homozygous with the dominant allele. All the code for data processing, model training, performance benchmarking, and model interpretation is available publicly at https://github.com/thepanlab/GattacaNet2.

4.2.2 Construction of the MTL and STL models.

The output of MTL LINA is a $d \times 1$ vector, Y, containing the predicted states of d traits. The input of MTL LINA is an $m \times 1$ vector, X, containing the genotypes of m SNPs. In this study, $d = 69$ in the pan-cancer MTL model, $d = 362$ in the pan-disease MTL model, and $m = 805426$ in both models. MTL LINA can be expressed as:

$$y = S(\mathbf{K} \cdot (A \odot X) + B) \tag{4.1}$$

$$A = F(X)$$

where $S()$ was a sigmoid activation function to be applied element-wise to its input column vector, $\mathbf{K}$ was a $d \times m$ coefficient matrix, A was a $m \times 1$ attention vector, B was a

$d \times 1$ bias vector, $\cdot$ represented the matrix-vector multiplication, and $\odot$ represented the element-wise multiplication. A was computed from X by a feedforward neural network, $F()$, composed of 3 hidden layers containing 1000, 250, and 50 neurons. A leaky-ReLU activation function, dropout with a dropout rate of 50%, and batch normalization were used in all three hidden layers. A linear activation function was used in the attention layer. $\mathbf{K}$, B, and $F()$ were all learned from the training data. The loss function of MTL LINA was defined as:

$$loss = W^T E + \beta ||\mathbf{K}||_2 \tag{4.2}$$

where W was a $d \times 1$ vector of the loss weights for all traits, E was a $d \times 1$ vector of the cross-entropy losses for all traits, and $||\mathbf{K}||_2$ was the L2 norm of the coefficient matrix, and β was the regularization weight. In this study, $W = [1, \ldots, 1]^T$ and $\beta = 10^{-3}$.
A total of 77 STL models were constructed for the 17 cancers and 60 non-cancer diseases with prevalence levels over 0.5%. All STL models used a feedforward neural network composed of three hidden layers containing 1000, 250, and 50 neurons as described previously (Badré et al., 2021). A leaky-ReLU activation function, dropout with a dropout rate of 50%, and batch normalization were also used in all three hidden layers. The cross-entropy loss function was used to train the STL models.

4.2.3 Training and benchmarking of the MTL and STL models

The 488,175 UKB subjects were randomly divided into a training set (70%), a validation set (15%), and a test set (15%). The training set was used to train all MTL and STL models by stochastic gradient descent. The training used mini-batches with a batch size of 512 and the Adam optimizer with an initial learning rate of 10^{-4}. All MTL and STL models were trained for 100 epochs with checkpointing after every epoch. The

checkpoints with the best performance on the validation set were kept for all MTL and STL models, which were the epoch-27 checkpoint for the pan-cancer MTL model and the epoch-25 checkpoint for the pan-disease MTL model. The training was carried out on a computer node with dual A100 40GB GPUs and 256 GB system memory. The training data was lazy-loaded to minimize memory usage using the pandas_plink (noa) library. After the training was completed, the predictive performances of all MTL and STL models were benchmarked using the test set.

4.2.4 Interpretation of the MTL models

The first-order model-wise LINA interpretation algorithm, as detailed in Equation 3.3 and the score FP (Equation 3.7), was used to identify important features (Badré and Pan, 2022) for each phenotype. A synthetic genomic vector was constructed for each subject to estimate the false discovery rate of the model interpretation, as shown previously (Badré and Pan, 2022). The synthetic genomic vectors of all subjects contained all their real SNPs and an equal number of decoy SNPs. The genotypes of the decoy SNPs were randomly set to be 0, 1, or 2 with the same probabilities observed in the real SNPs. Thus, the decoy SNPs had identical frequencies of homozygous minor alleles, heterozygous alleles, and homozygous dominant alleles as the real SNPs. But, because the decoy SNPs should have no association with the phenotypes, any decoy SNP identified as important by the interpretation algorithm was considered a false positive hit.

A pan-cancer MTL model was constructed and trained as described above using the synthetic genomic vectors of the subjects in the training set. The importance scores of both real and decoy SNPs were computed for each cancer using the subjects in the test set. Only SNPs on the non-sex chromosomes were considered for model interpretation. The FDR for an importance score threshold was estimated as the ratio between

the numbers of decoy SNPs to real SNPs above this threshold. The important SNPs at 0.1% FDR and 5% FDR were identified for all cancers with $> 0.5\%$ prevalence in the pan-cancer MTL model. The intersection and union of the important SNPs were counted between every pair of prevalent cancers. The genetic correlation between two cancers was computed as the Spearman correlation coefficient between the importance scores of the SNPs belonging to the union of the SNP sets of the two cancers at 5% FDR.

4.3 Results

4.3.1 Parallel prediction of many diseases by MTL

A neural network architecture was developed to predict many traits of an individual from their whole genome (Figure 4.1). This was an MTL extension of the linearizing neural network architecture (LINA) previously shown to provide good predictive performance for STL estimation of breast cancer PRS (Badré and Pan, 2022). All the traits shared a latent genomic representation, which was an element-wise multiplication of a learned attention vector and the input whole-genome vector. Each trait was predicted from the shared representation via a task-specific hidden layer. The output from the MTL model was a vector of character states for all the considered phenotypes, referred to as a whole-phenome vector.

Training a MTL model required a cohort of subjects with phenome-wide trait data. In this study, we used the United Kingdom Biobank (UKB) dataset and extracted 362 disease traits, including 69 cancer traits, from the electronic medical record of 488,175 UKB participants. 77 diseases, including 17 types of cancers and 60 non-cancer diseases, had prevalence levels higher than 0.5% in the UKB cohort. We constructed two

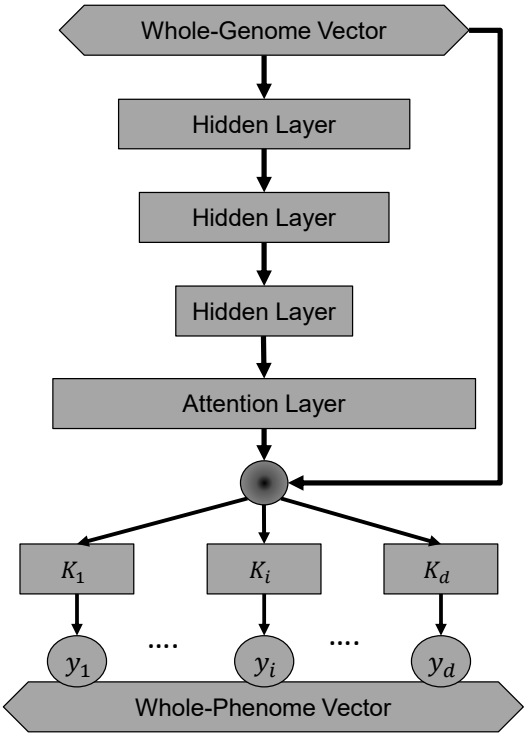


Figure 4.1: **An MTL deep neural network for parallel prediction of multiple traits.** This model was constructed based on the linearizing neural network architecture. The input layer (diamond box) contains all genetic variants in the whole genome. An attention vector is generated after 3 hidden layers (rectangular boxes) and then multiplied element-wise (round circle) with the input vector through a skip connection. The shared representation is used to predict each trait (y_i in round circle). From end to end, a whole-phenome vector (diamond box) composed of many individual traits is predicted from this individual's whole-genome vector.

MTL models, one to predict the 69 cancers (pan-cancer MTL) and the other one to predict all 362 diseases (pan-disease MTL). Instead of selecting SNPs for each disease based on their statistical association, we included all 805,426 SNPs genotyped in the UKB cohort as the input for both MTL models. The UKB cohort was randomly divided into a training set (70%) for model training, a validation set (15%) for hyperparameter optimization, and a test set (15%) for performance benchmarking. A model's training took approximately 5 days on a computer node with dual A100 40GB GPUs. All the benchmarking results described below were based on the test set.

4.3.2 Improved accuracy for PRS estimation by MTL

The estimation accuracy of malignant melanoma PRS was compared among STL, pan-cancer MTL, and pan-disease MTL (Figure 4.2 and Figure 4.3). The same training data was used to train the STL model to predict malignant melanoma only (Figure 4.2A), the pan-cancer MTL model to predict 69 cancers, including malignant melanoma (Figure 4.2B), and the pan-disease MTL to predict malignant melanoma along with 361 other diseases (Figure 4.2C). The MTL and STL models generated different distributions of PRS in the test set. The differences were especially pronounced on the two shoulders of the distributions, which represented the subjects with higher or lower genetic risks than the average. The separation between the PRS distribution of the control subjects and the PRS distribution of the case subjects was greater in the two MTL models than the STL model. The predictive performances of the three models were compared using the Receiver Operating Characteristics (ROC) curves (Figure 4.3A). The Area Under the Curve (AUC) of the ROC curve by STL was only 2.8% higher than the 50% baseline, while those by the pan-cancer MTL and pan-disease MTL were 9.2% and 8.1% higher, respectively. Because of the imbalanced data, the predictive performances of

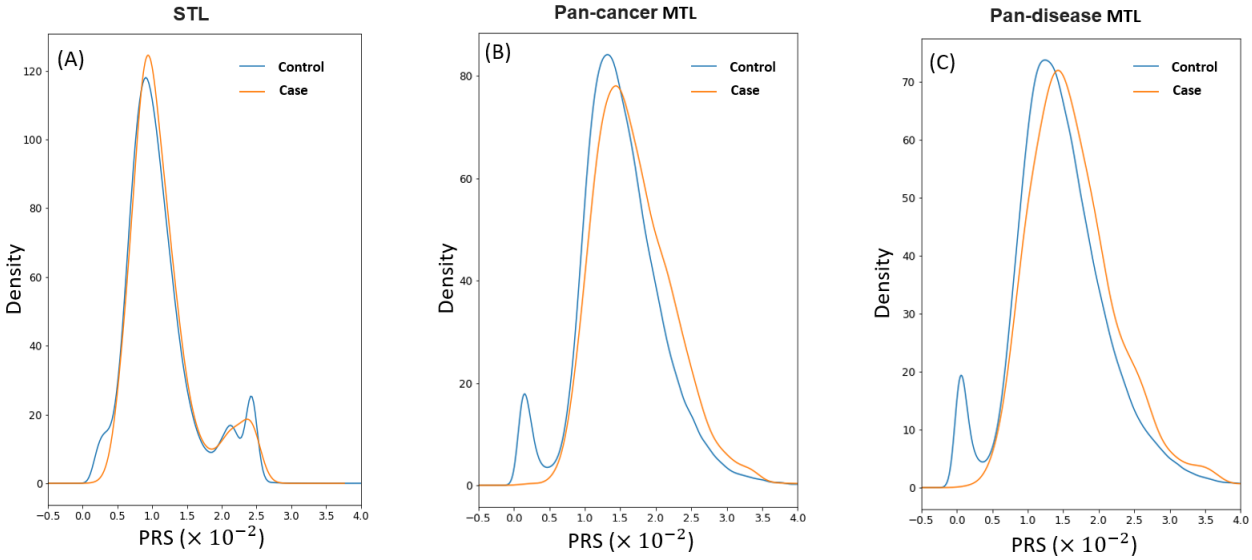


Figure 4.2: **PRS estimation for malignant melanoma by STL and MTL.** (A – C) Density plots of malignment melanoma PRS estimated by (A) STL, (B) pan-cancer MTL, and (C) pan-disease MTL. Each panel contains two overlapping density plots: a blue one for the control test cohort and an orange one for the case test cohort. The separation between the control and case density plots is greater in the two MTL panels than in the STL panel.

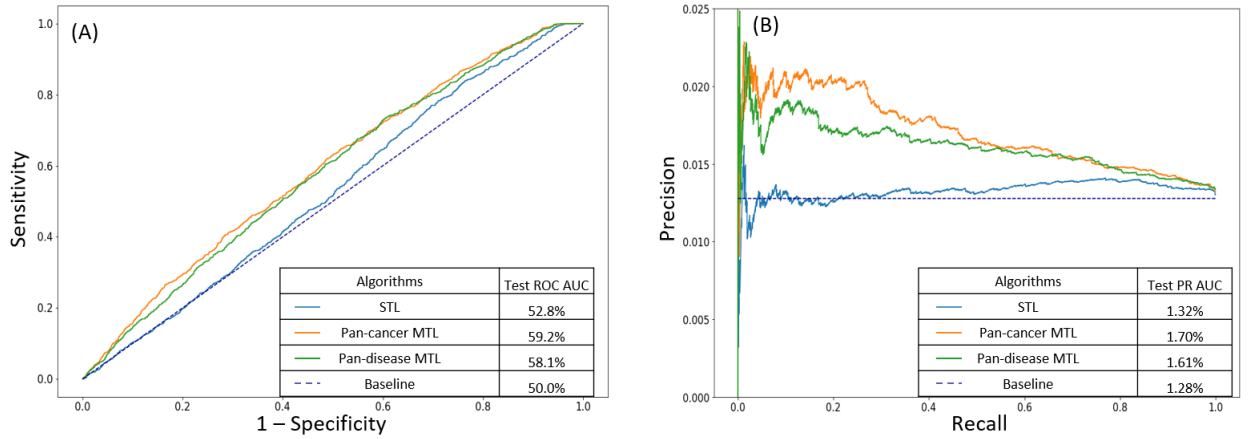


Figure 4.3: **PRS ROC AUC and PR AUC curves for malignant melanoma by STL and MTL.** (A) Receiver operating characteristic (ROC) curves of STL (blue), pan-cancer MTL (orange), and pan-disease MTL (green) for malignant melanoma PRS with the baseline (indigo dotted line). Both pan-cancer MTL and pan-disease MTL have larger ROC AUC than STL. (B) Precision-recall (PR) curves of STL (blue), pan-cancer MTL (orange), and pan-disease MTL (green) for malignant melanoma PRS with the disease prevalence as the baseline (indigo dotted line). The two MTL models also have larger PR AUC than STL.

the three models were also compared using the Precision-Recall (PR) curves (Figure 4.3B). Both MTL models achieved much higher precisions at the same recall level than the STL model. The baseline of the PR curve was determined by the prevalence level of malignant melanoma in the UKB cohort, which was 1.28%. The PR AUC by STL was 0.04% higher than the baseline, while those by the pan-cancer MTL and pan-disease MTL were 0.42% and 0.33% higher, respectively. Overall, these metrics reflected better predictive performance of the two MTL models than the STL model for malignant melanoma.

The predictive performances of the two MTL models were then compared with the disease-specific STL models across 17 common cancers with prevalence levels higher than 0.5% (Table 4.1). The comparisons were made using both ROC AUC and PR AUC to account for the sensitivity, specificity, precision, and recall of the models. The two MTL models offered higher ROC AUC for 16 cancers and higher PR AUC for all 17 cancers than the disease-specific STL models. The magnitude of the performance improvement was quantified using the relative increase of the over-the-baseline AUC gain by an MTL model in comparison with the corresponding STL model. The average relative increase of ROC AUC over STL was 141% for the pan-cancer MTL and 153% for the pan-disease MTL. The average relative increase of PR AUC over STL was 96% for the pan-cancer MTL and 83% for the pan-disease MTL. The variability of the relative increases among different cancers suggested that each disease benefited to a different extent from MTL. The pan-cancer MTL had the highest ROC AUC for 4 cancers and highest PR AUC for 5 cancers. The pan-disease MTL had the highest ROC AUC for 12 cancer types and highest PR AUC for 12 cancer types. This suggested that the performance improvement from transfer learning increased with the number of traits in MTL. To further check if the performance gain by MTL over STL can

Diseases	Receiver operating characteristics (ROC) AUC#					Precision-recall (PR) AUC#					
	STL ROC AUC	Pan-cancer MTL		Pan-disease MTL		STL PR AUC	Pan-cancer MTL		Pan-disease MTL		Prevalence (Baseline)
		ROC AUC	Relative increase*	ROC AUC	Relative increase*		PR AUC	Relative increase*	PR AUC	Relative increase*	
Malignant melanoma	52.80%	**59.20%**	**234%**	58.10%	194%	1.33%	**1.70%**	**790%**	1.61%	593%	1.28%
Non-melanoma skin cancer	61.50%	62.40%	8%	**62.90%**	**12%**	9.76%	10.03%	8%	**10.21%**	**14%**	6.65%
Skin cancer	61.00%	61.80%	7%	**61.90%**	**8%**	10.40%	10.73%	11%	**10.86%**	**15%**	7.32%
Lung cancer	59.10%	60.30%	14%	**60.50%**	**16%**	1.39%	1.44%	11%	**1.51%**	**23%**	0.90%
Intrathoracic cancer	59.10%	60.70%	18%	**61.00%**	**21%**	1.54%	1.58%	7%	**1.65%**	**20%**	1.01%
Colorectal cancer	54.40%	56.40%	46%	**57.10%**	**60%**	2.00%	2.21%	71%	**2.29%**	**100%**	1.72%
Colon cancer	53.90%	55.70%	47%	**56.10%**	**59%**	1.38%	1.49%	51%	**1.49%**	**53%**	1.17%
Rectal cancer	54.70%	57.90%	69%	**59.30%**	**100%**	0.77%	0.88%	98%	**0.89%**	**116%**	0.67%
Bladder cancer	64.50%	67.90%	24%	**68.40%**	**27%**	0.80%	0.87%	24%	**0.92%**	**42%**	0.51%
Uterine cancer	51.20%	**53.20%**	**177%**	51.80%	50%	1.08%	**1.18%**	**224%**	1.10%	49%	1.04%
Cervical cancer	55.20%	55.40%	4%	**56.50%**	**24%**	1.80%	1.88%	35%	**1.97%**	**76%**	1.58%
Prostate cancer	**60.00%**	59.70%	-3%	59.60%	-4%	8.33%	**8.53%**	**9%**	8.37%	2%	6.06%
Breast cancer	57.00%	**58.30%**	**19%**	58.10%	16%	9.38%	9.67%	13%	**9.79%**	**20%**	7.25%
Female genital tract cancer	54.00%	54.30%	7%	**54.50%**	**11%**	3.15%	3.27%	41%	**3.39%**	**84%**	2.86%
Male genital tract cancer	53.60%	**56.10%**	**68%**	54.50%	24%	2.57%	**2.73%**	**57%**	2.59%	9%	2.28%
Lymphoma	50.40%	56.80%	1442%	**57.90%**	**1704%**	0.73%	**0.82%**	**102%**	0.82%	98%	0.64%
Non-hodgkins lymphoma	52.10%	56.60%	220%	**57.80%**	**278%**	0.61%	0.68%	81%	**0.69%**	**95%**	0.53%

#Best AUC highlighted in bold

*Relative increase = $\frac{(MTL\ AUC - baseline\ AUC) - (STL\ AUC - baseline\ AUC)}{STL\ AUC - baseline\ AUC} \times 100\%$

Table 4.1: **Comparison of STL, pan-cancer MTL, and pan-disease MTL by ROC AUC and PR AUC for 17 cancer types with** $> 0.5\%$ **prevalence**

Disease	Receiver operating characteristics (ROC) AUC#			Precision-recall (PR) AUC#			
	STL	Pan-disease MTL ROC		STL	Pan-disease MTL PR		Prevalence (Baseline)
		ROC AUC	Relative increase*		PR AUC	Relative increase*	
Allergy or anaphylactic reaction to food	54.35%	**55.52%**	**26.80%**	0.61%	**0.64%**	**44.67%**	0.54%
Angina	64.08%	**65.33%**	**8.88%**	5.52%	**5.60%**	**3.59%**	3.23%
Appendicitis	51.16%	**53.10%**	**167.60%**	1.44%	**1.47%**	**29.03%**	1.32%
Arthritis (nos)	55.17%	**55.64%**	**9.02%**	1.51%	**1.52%**	**6.25%**	1.22%
Asthma	60.01%	**61.15%**	**11.39%**	16.44%	**17.02%**	**12.37%**	11.75%
Atrial fibrillation	63.69%	**64.62%**	**6.78%**	4.12%	**4.42%**	**18.82%**	2.50%
Atrial flutter	64.05%	**64.69%**	**4.59%**	3.60%	**3.70%**	**6.62%**	2.12%
Back pain	52.75%	**53.44%**	**24.94%**	1.15%	**1.19%**	**34.87%**	1.03%
Benign breast lump	53.39%	**54.18%**	**23.45%**	1.58%	**1.62%**	**23.64%**	1.41%
Bronchitis	51.87%	**52.95%**	**57.92%**	1.32%	**1.33%**	**9.67%**	1.21%
Cataract	52.81%	**53.43%**	**22.13%**	2.38%	**2.45%**	**29.45%**	2.13%
Cervical spondylosis	56.00%	**58.97%**	**49.47%**	0.87%	**1.02%**	**95.82%**	0.72%
Chickenpox	53.73%	**54.16%**	**11.41%**	2.58%	**2.62%**	**20.03%**	2.33%
Gallstones	63.43%	**64.95%**	**11.36%**	2.65%	**2.86%**	**23.82%**	1.75%
Chronic sinusitis	53.23%	**53.92%**	**21.31%**	0.82%	**1.07%**	**343.92%**	0.75%
Depression	57.35%	**58.46%**	**15.15%**	7.93%	**8.29%**	**20.87%**	6.24%
Diabetes	67.26%	**67.92%**	**3.83%**	9.75%	**9.99%**	**4.78%**	4.59%
Dyspepsia / indigestion	53.67%	**55.05%**	**37.66%**	3.13%	**3.34%**	**64.05%**	2.80%
Ear/vestibular disorder	51.57%	**53.16%**	**101.20%**	**1.01%**	1.00%	-7.92%	0.92%
Eczema/dermatitis	54.62%	**56.98%**	**51.32%**	3.54%	**3.93%**	**80.88%**	3.07%
Emphysema/chronic bronchitis	58.26%	**60.08%**	**22.08%**	1.94%	**2.03%**	**14.61%**	1.35%
Endometriosis	**54.13%**	53.80%	-7.88%	**1.77%**	1.75%	-9.67%	1.58%
Epilepsy	49.81%	**53.94%**	**2188.46%**	0.81%	**0.93%**	**1638.72%**	0.82%
Essential hypertension	51.92%	**54.37%**	**127.01%**	1.38%	**1.47%**	**78.20%**	1.26%
Eye/eyelid problem	49.52%	**50.96%**	**298.50%**	0.79%	**0.84%**	**711.13%**	0.80%
Fracture lower leg / ankle	53.99%	**54.65%**	**16.63%**	0.56%	**0.60%**	**89.87%**	0.52%
Gastro-oesophageal reflux (gord) / gastric reflux	53.91%	**55.18%**	**32.31%**	5.57%	**5.73%**	**22.40%**	4.87%
Glaucoma	58.48%	**59.31%**	**9.86%**	1.65%	**1.83%**	**42.36%**	1.24%
Gout	61.60%	**63.60%**	**17.21%**	4.66%	**5.32%**	**43.41%**	3.16%
Hayfever/allergic rhinitis	55.02%	**57.16%**	**42.54%**	7.62%	**8.13%**	**44.51%**	6.48%
Heart attack (MI)	71.25%	**72.56%**	**6.16%**	5.06%	**5.27%**	**7.30%**	2.30%
Hiatus hernia	55.90%	**57.71%**	**30.64%**	3.06%	**3.26%**	**36.16%**	2.51%
Hypertension	**62.89%**	61.78%	-8.64%	**37.70%**	36.41%	-12.40%	27.32%
Hyperthyroidism/thyrotoxicosis	67.77%	**71.55%**	**21.27%**	1.74%	**2.14%**	**45.39%**	0.85%
Hypothyroidism/myxoedema	73.90%	**74.96%**	**4.43%**	12.99%	**13.59%**	**7.53%**	5.14%
Inguinal hernia	75.09%	**76.31%**	**4.89%**	1.49%	**1.57%**	**9.41%**	0.66%
Iron deficiency anaemia	67.23%	**67.34%**	**0.65%**	1.53%	**1.61%**	**10.74%**	0.75%
Irritable bowel syndrome	62.16%	**62.38%**	**1.78%**	**3.85%**	3.81%	-2.77%	2.56%
Joint pain	50.89%	**51.78%**	**100.28%**	0.61%	**0.69%**	**468.36%**	0.59%
Kidney stone/ureter stone/bladder stone	60.69%	**65.07%**	**41.01%**	1.29%	**1.53%**	**52.22%**	0.84%
Measles / morbillivirus	**54.81%**	53.73%	-22.39%	**2.06%**	1.97%	-34.87%	1.78%
Migraine	64.51%	**64.65%**	**0.99%**	5.48%	**5.57%**	**3.91%**	3.38%
Mumps / epidemic parotitis	55.02%	**55.73%**	**14.19%**	1.23%	**1.33%**	**85.88%**	1.11%
Oesophagitis/barretts oesophagus	54.94%	**55.57%**	**12.66%**	2.57%	**2.60%**	**6.79%**	2.16%
Osteoporosis	71.77%	**71.94%**	**0.77%**	**3.67%**	3.65%	-1.38%	1.76%
Other joint disorder	50.51%	**50.55%**	**8.05%**	**0.62%**	0.61%	-27.80%	0.60%
Pleural effusion	55.53%	**55.92%**	**7.08%**	**0.78%**	0.77%	-2.67%	0.61%
Pneumonia	52.46%	**53.75%**	**52.23%**	1.89%	**1.98%**	**71.05%**	1.76%
Psoriasis	65.93%	**69.96%**	**25.32%**	2.73%	**3.60%**	**59.80%**	1.26%
PE +/- DVT	53.52%	**54.39%**	**24.61%**	**1.12%**	1.04%	-33.39%	0.88%
Respiratory infection	53.23%	**55.10%**	**58.13%**	2.15%	**2.21%**	**22.05%**	1.88%
Rheumatoid arthritis	60.03%	**62.85%**	**28.14%**	1.79%	**2.21%**	**77.95%**	1.25%
Rubella / german measles	53.95%	**54.71%**	**19.12%**	0.74%	**0.82%**	**106.25%**	0.68%
Sciatica	52.14%	**53.54%**	**65.85%**	1.25%	**1.33%**	**65.47%**	1.14%
Stroke	59.06%	**59.80%**	**8.11%**	2.14%	**2.27%**	**23.21%**	1.58%
Tuberculosis (tb)	51.42%	**53.69%**	**160.91%**	0.57%	**0.68%**	**307.11%**	0.54%
Ulcerative colitis	57.43%	**60.72%**	**44.32%**	0.77%	**0.77%**	**4.13%**	0.55%
Urinary tract infection/kidney infection	**58.20%**	58.19%	-0.17%	**0.76%**	0.75%	-5.71%	0.58%
Varicose veins	58.09%	**59.84%**	**21.72%**	0.71%	**0.76%**	**27.42%**	0.52%
Whooping cough / pertussis	**52.32%**	51.15%	-50.68%	0.83%	**0.84%**	**12.82%**	0.79%

#Best AUC highlighted in bold

*Relative increase = $\frac{(MTL\ AUC - baseline\ AUC) - (STL\ AUC - baseline\ AUC)}{STL\ AUC - baseline\ AUC} \times 100\%$

Table 4.2: **Comparison of STL and pan-disease MTL by ROC AUC and PR AUC for 60 non-cancer diseases with $> 0.5\%$ prevalence.**

be generalized across non-cancer diseases, we compared the pan-disease MTL model with the disease-specific STL models for 60 non-cancer diseases with prevalence levels higher than 0.5% (Table 4.2). The same set of performance metrics was used for the comparison. Compared with the disease-specific STL models, the pan-disease MTL model provided higher ROC AUC for 55 non-cancer diseases and higher PR AUC for 50 non-cancer diseases. The average relative increase by MTL across the 60 non-cancer diseases was 68% for ROC AUC and 82% for PR AUC. The benchmarking results for both cancer and non-cancer diseases indicated significant performance improvements by MTL over STL across many diseases.

4.3.3 Identification of important SNPs for MTL by model interpretation

The first-order model-wise LINA interpretation algorithm (see Chapter 3) was used to identify the important SNPs used by MTL to predict each disease. A pan-cancer MTL model was trained and interpreted using an input whole-genome vector that contained the real SNPs and an equal number of decoy SNPs. Figure 4.4 shows the distributions of importance scores for the real SNPs and the decoy SNPs used by the MTL model to predict malignant melanoma. There were 59,350 real SNPs and 3091 decoy SNP with important scores above 0.52×10^{-3}, which corresponded to a 5% FDR, because decoy SNPs with random association with the trait cannot be truly important for prediction. At the estimated FDR level of 0.1%, 48 real SNPs and no decoy SNP were identified as important for the MTL model to predict malignant melanoma. Many of these important SNPs have been identified as risk variants for melanoma in previous GWAS studies, including rs12203592 (Gibbs et al., 2017), rs62389423 (Ransohoff et al., 2017), rs4785763 (Bishop et al., 2009), rs4238833 (Bishop et al., 2009), rs10931936 (Landi

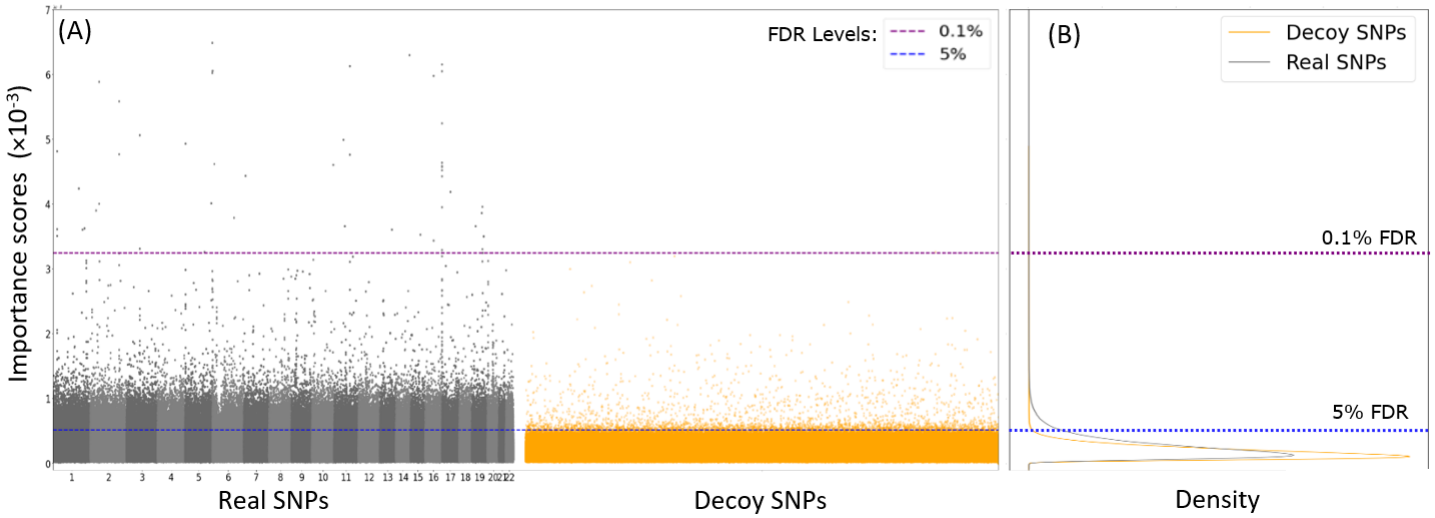


Figure 4.4: **Importance scores of real and decoy SNPs for malignant melanoma PRS estimation by pan-cancer MTL.** (A) Manhattan plots of real SNPs (black and grey dots) and decoy SNPs (orange dots) by their importance scores. (B) density plots of the importance scores of real SNPs (black curve) and decoy SNPs (orange curve). An estimated FDR of 5% (3091 decoy SNP to 59,350 real SNPs) was reached at the importance score threshold of 0.52×10^3 (blue dotted line). No decoy SNPs and 48 real SNPs have importance scores above the threshold of 3.25×10^3 for an estimated 0.1% FDR (purple dotted line).

et al., 2020), rs1126809 (Landi et al., 2020), and Affx-35293625 (Brandes et al., 2021).

Important SNPs in the pan-cancer MTL model were identified for the 17 prevalent cancers at the FDR levels of 0.1% and 5% (Table 4.3). The number of important SNPs at 0.1% FDR was 29 on average across the 17 cancers with substantial variability. These important SNPs may have strong associations with the traits. At 5% FDR, an average of 36,048 important SNPs were identified for the cancers, suggesting the use of diffused weak association signals across the whole genome by MTL for trait prediction. We investigated the overlaps among the important SNPs for different diseases. At 0.1% FDR, only 4 common SNPs were shared among uterine cancer's 25 important SNPs, colorectal cancer's 36 important SNPs, and malignant melanoma's 48 important SNPs (Figure 4.5A). The number of important SNPs in the intersection for every pair of diseases at 0.1% FDR were listed in Table 4.4. The relatively small intersections between different cancers indicated distinct SNP sets with large effect sizes for different diseases. At 5% FDR, there were 21041 common SNPs shared among uterine cancer's 38474 important SNPs, colorectal cancer's 45450 important SNPs, and malignant melanoma's 59350 important SNPs (Figure 4.5B). Genetic correlations were computed between every pair of cancers based on their importance scores for the SNPs important for one of the diseases or both at 5% FDR (Table 4.5). The genetic correlations were 0.88 between breast cancer and uterine cancer and 0.89 between lung cancer and lymphoma. Overall, 184 pairs of diseases have positive correlation coefficients between 0.5 and 1.0, 97 pairs have positive correlation coefficients between 0 and 0.5, and only 8 pairs have negative correlation coefficients. This suggested that MTL identified and

FDR levels / Disease	0.1%	5.0%
Malignant melanoma	48	59350
Non-melanoma skin cancer	132	48848
Skin cancer	106	48419
Lung cancer	4	41075
Intrathoracic cancer	3	40392
Colorectal cancer	36	45450
Colon cancer	22	37487
Rectal cancer	28	47904
Bladder cancer	8	37
Uterine cancer	25	38474
Cervical cancer	5	42068
Prostate cancer	23	94
Breast cancer	34	96
Female genital tract cancer	15	37083
Male genital tract cancer	5	40742
Lymphoma	0	43412
Non-hodgkins lymphoma	4	41889

Table 4.3: **Numbers of important SNPs used by pan-cancer MTL to estimate PRS of prevalent cancers**

Malignant melanoma	Non-melanoma skin cancer	Skin cancer	Lung cancer	Intrathoracic cancer	Colorectal cancer	Colon cancer	Rectal cancer	Bladder cancer	Uterine cancer	Cervical cancer	Prostate cancer	Breast cancer	Female genital tract cancer	Male genital tract cancer	Lymphoma	Non-hodgkins lymphoma	
	32	35	1	1	10	7	13	2	6	0	6	12	4	2	0	1	Malignant melanoma
		102	3	2	18	9	16	4	11	2	7	17	7	2	0	2	Non-melanoma skin cancer
			3	2	18	9	16	3	10	1	8	13	6	2	0	1	Skin cancer
				3	1	1	2	0	1	1	1	1	0	0	0	1	Lung cancer
					1	1	2	0	1	1	1	1	0	0	0	1	Intrathoracic cancer
						16	18	3	8	1	7	5	4	3	0	2	Colorectal cancer
							6	1	6	1	5	1	3	1	0	3	Colon cancer
								4	8	1	5	7	4	2	0	2	Rectal cancer
									2	0	1	0	0	0	0	0	Bladder cancer
										1	3	4	5	2	0	2	Uterine cancer
											1	0	3	1	0	1	Cervical cancer
												5	2	3	0	0	Prostate cancer
													2	1	0	0	Breast cancer
														2	0	1	Female genital tract cancer
															0	0	Male genital tract cancer
																0	Lymphoma
																	Non-hodgkins lymphoma

Table 4.4: **Numbers of shared important SNPs at 0.1% FDR between prevalent cancers in pan-cancer MTL.**

Malignant melanoma	Non-melanoma skin cancer	Skin cancer	Lung cancer	Intrathoracic cancer	Colorectal cancer	Colon cancer	Rectal cancer	Bladder cancer	Uterine cancer	Cervical cancer	Prostate cancer	Breast cancer	Female genital tract cancer	Male genital tract cancer	Lymphoma	Non-hodgkins lymphoma	
	0.72	0.73	0.3	0.3	0.4	0.27	0.41	0.42	0.32	0.39	0.5	0.57	0.26	0.39	0.28	0.25	Malignant melanoma
		0.96	0.09	0.09	0.15	0.03	0.16	0.4	0.05	0.15	0.44	0.49	0.04	0.12	0.04	0.02	Non-melanoma skin cancer
			0.05	0.05	0.12	0	0.13	0.38	0.03	0.11	0.42	0.48	0	0.09	0.01	-0.01	Skin cancer
				0.99	0.86	0.86	0.86	0.74	0.78	0.84	0.91	0.87	0.86	0.8	0.89	0.89	Lung cancer
					0.85	0.84	0.85	0.76	0.75	0.83	0.91	0.86	0.86	0.78	0.88	0.87	Intrathoracic cancer
						0.9	0.93	0.65	0.81	0.87	0.84	0.87	0.81	0.86	0.86	0.84	Colorectal cancer
							0.83	0.69	0.83	0.81	0.87	0.87	0.85	0.81	0.9	0.9	Colon cancer
								0.66	0.82	0.88	0.85	0.86	0.81	0.87	0.86	0.84	Rectal cancer
									0.6	0.65	-0.43	-0.36	0.74	0.6	0.69	0.71	Bladder cancer
										0.8	0.77	0.88	0.78	0.82	0.85	0.84	Uterine cancer
											0.82	0.89	0.86	0.81	0.84	0.82	Cervical cancer
												-0.52	0.88	0.82	0.89	0.9	Prostate cancer
													0.88	0.82	0.89	0.88	Breast cancer
														0.75	0.87	0.88	Female genital tract cancer
															0.85	0.82	Male genital tract cancer
																0.98	Lymphoma
																	Non-hodgkins lymphoma

Table 4.5: **Genetic correlations between every pair of prevalent cancers in pan-cancer MTL.** The correlation coefficients are computed between the importance scores of the SNPs important for both or one of the two cancers at 5% FDR. The importance scores of many pairs of cancers have high correlation coefficients, which indicate shared genetic basis.

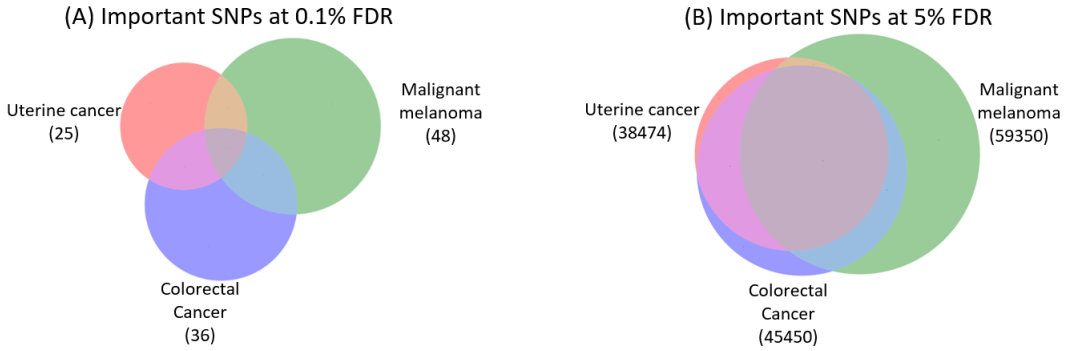


Figure 4.5: **Importance scores of real and decoy SNPs for malignant melanoma PRS estimation by pan-cancer MTL.** The Venn diagrams show the overlap among the important SNPs found for uterine cancer, malignant melanoma, and colorectal cancer at 0.1% FDR (A) and 5% FDR (B). The sizes of the circles and their overlaps are drawn proportionally. The important sets of SNPs for the three cancers have a small overlap at 0.1% FDR and a large overlap at 5% FDR.

may have exploited extensive genetic correlations between diseases to achieve a positive knowledge transfer among diseases for PRS estimation.

4.4 Discussion

Learning many tasks together in a neural network model does not automatically guarantee performance boost for all tasks (Fifty et al., 2021), (Joshi et al., 2019). Negative knowledge transfer can occur between unrelated tasks and, thereby, degrade the performance of a MTL model for these tasks (Bingel and Søgaard, 2017). We did not assume a priori which sets of diseases might be genetically related and could benefit from MTL. By aggregating many diseases together, we discovered positive knowledge transfer for most of the prevalent diseases studied here. The extent of positive knowledge transfer was quantified for each disease based on the gain of predictive performance by MTL relative to STL. For example, malignment melanoma and uterine cancer benefited substantially from parallel training with the other cancers in the pan-cancer MTL, but the extent of positive knowledge transfer to the two cancers was reduced when adding

many non-cancer diseases in the pan-disease MTL. The majority of common cancers, including intrathoracic cancer, rectal cancer, and cervical cancer, gained additional performance by scaling MTL from 69 cancers to 362 diseases. Beneficial transfer learning was also evident for most of the non-cancer common diseases. Consistent observation of increased PRS accuracies for so many diseases provided strong support for the positive knowledge transfer during parallel learning of the genetic risks for complex diseases.

To understand how the PRS estimation benefited from MTL, we interpreted a pan-cancer MTL model and identified important SNPs for each cancer at two empirically estimated FDR levels. Many diseases shared a significant fraction of important SNPs at 5% FDR for their predictions. This suggested a beneficial joint selection of SNPs predictive of multiple diseases. This could be attributed to pleiotropy, wherein a genetic variant may have effects on multiple traits. A meta-analysis of many complex traits' GWAS results estimated 31% of the SNPs and 63% of the genes to be pleiotropic (Watanabe et al., 2019). In addition, the joint feature selection in MTL may be better at filtering out SNPs with random trait associations in the training data than the disease-specific feature selection in STL can.

Data amplification may be a second mechanism for beneficial transfer learning in PRS estimation. Many diseases have an epidemiological correlation. For example, Woo et al. found a 75% greater risk of overall incident cancers after asthma diagnosis in adults (Woo et al., 2021). Pooling the positive cases of multiple diseases together to train a MTL model may increase the effective sample size for learning a shared latent representation predictive of these diseases. Furthermore, many cancers may have some common genetic etiology. Pan-cancer risk variants may elevate the overall risk of individuals for cancers (Rashkin et al., 2020), and some environmental factors may determine the specific site of carcinogenesis. Pooling many cancer cases together may amplify the signal for discovering pan-cancer risk variants. Besides feature selection

and data amplification, other mechanisms, such as eavesdropping, representation bias, and regularization (Caruana, 1998), may also contribute to the positive knowledge transfer between diseases for PRS estimation.

Because hard parameter sharing was used in our neural networks from the input layer to the attention layer, the beneficial transfer learning may have produced a latent representation of the genomic data with better generalization for many diseases. Pervasive genetic correlations between diseases allowed MTL to improve the PRS estimation broadly across diseases. While many cross-strait studies have shown the genetic correlation between specific pairs of diseases (Zhuang et al., 2021; Sutton et al., 2022; Wu et al., 2022; Zhou et al., 2022; Adewuyi et al., 2022; Lutz et al., 2020; Byun et al., 2021), our study suggested that various degrees of shared genetic basis may be very prevalent among many complex diseases. Our results highlighted the potential value of holistic association studies between the whole human phenome and the whole human genome for both risk variant discovery and PRS estimation.

Chapter 5

Summary and Conclusions

5.1 Deep neural network improves the estimation of polygenic risk scores for breast cancer

This study compared different computational models for estimating polygenic risk scores (PRS) for breast cancer using genetic variants across the whole genome. A deep neural network (DNN) outperformed established statistical algorithms such as BLUP, BayesA, and LDpred. In a test cohort with 50% prevalence, DNN achieved an area under the receiver operating characteristic curve (AUC) of 67.4% and was able to separate the case population into high- and normal-genetic-risk sub-populations. The PRS generated by DNN in the case population followed a bi-modal distribution composed of two normal distributions with distinctly different means. This suggests that DNN was able to separate the case population into a high-genetic-risk case sub-population with an average PRS significantly higher than the control population and a normal-genetic-risk case sub-population with an average PRS similar to the control population. This allowed DNN to achieve 18.8% recall at 90% precision in the test cohort with 50% prevalence, which can be extrapolated to 65.4% recall at 20% precision in a general population with 12% prevalence. Interpretation of the DNN model identified interesting variants assigned insignificant p-values by association studies but were important for DNN prediction. These variants may be associated with the phenotype through non-linear relationships or epistatic interactions.

This study, however, presents some limitations. First, we didn't restrict the study to one ancestry group, which could lead to biased models. Nevertheless, mitochondrial SNPs were included in the study, and the models were aware of ancestry to some extent. We assumed that any good model could exploit linkage disequilibrium and HWE disequilibrium. Thus, we didn't perform classic preprocessing steps, which could hinder the performances of benchmark models. However, these choices were critical since

we needed to relax the proposed criteria to allow non-linear relationship expressions. Future works for this study could focus on studying those criteria in depth.

This paper was published in "Badré, A., Zhang, L., Muchero, W. et al. Deep neural network improves the estimation of polygenic risk scores for breast cancer. J Hum Genet 66, 359–369 (2021). https://doi.org/10.1038/s10038-020-00832-7"

5.2 LINA: A Linearizing Neural Network Architecture for Accurate First-Order and Second-Order Interpretations

Although neural networks can yield high predictive performance, the lack of interpretability has hindered the identification of salient features and important feature interactions used for their predictions. This represented a key hurdle for deploying neural networks in many biomedical applications that require interpretability, including predictive genomics. LINA was developed to provide both the first-order and the second-order interpretations on both the instance-wise and the model-wise levels. LINA combines the representational capacity of a deep inner attention neural network with a linearized intermediate representation for model interpretation. In comparison with DeepLIFT, LIME, Grad*Input, and L2X, the first-order interpretation of LINA had better Spearman correlations with the ground-truth importance rankings of features in synthetic datasets. In comparison with NID and GEH, the second-order interpretation results from LINA achieved better precision for the identification of the ground-truth feature interactions in synthetic datasets. These algorithms were further benchmarked using predictive genomics as a real-world application. LINA identified larger numbers

of SNPs and salient SNP interactions than the other algorithms at given false discovery rates. The results showed accurate and versatile model interpretation using LINA.

In this paper, some limitations can be highlighted again. First, the genomics part was reduced to only 3082 SNPs. In larger studies, as in Section 2, the total number of SNPs considered at the optimal threshold was 5273, or more than 1M in Section 4. It would have been more appropriate to study the behavior of LINA on a higher number of variants. Nevertheless, standard GWAS also works with a limited number of SNPs because of the restrictive selection threshold.

This paper was published in "A. Badré and C. Pan, "LINA: A Linearizing Neural Network Architecture for Accurate First-Order and Second-Order Interpretations," in IEEE Access, vol. 10, pp. 36166-36176, 2022, doi: 10.1109/ACCESS.2022.3163257."

5.3 Explainable multi-task learning improves the parallel estimation of polygenic risk scores for many diseases through shared genetic basis

In this study, we developed a multi-task learning (MTL) neural network architecture to predict many disease traits of an individual from their whole genome. The model used a shared latent genomic representation, and each trait was predicted from the shared representation via a task-specific hidden layer. The study used the UK Biobank dataset to extract 362 disease traits, including 69 cancer traits and constructed two MTL models - one to predict the 69 cancers and the other to predict all 362 diseases. The MTL models achieved higher predictive performance than single-task learning (STL) models for malignant melanoma and 17 common cancers with prevalence levels higher

than 0.5%. The MTL models also showed improved accuracy for predicting 60 non-cancer diseases with prevalence levels higher than 0.5%. The study suggested that the performance improvement from transfer learning increased with the number of traits in MTL. The first-order model-wise LINA interpretation algorithm was utilized to identify important SNPs used by Multi-Task Learning (MTL) to predict cancer diseases. A pan-cancer MTL model was trained and interpreted using real and decoy SNPs. At FDR levels of 0.1% and 5%, important SNPs were identified for 17 prevalent cancers, with a higher number of important SNPs identified at 5% FDR. The overlaps among the important SNPs for different diseases were investigated, and small intersections between different cancers were found, indicating distinct SNP sets with large effect sizes for different diseases. At 5% FDR, genetic correlations were computed between every pair of cancers based on their importance scores for the SNPs important for one of the diseases or both. The genetic correlations suggested that MTL identified and exploited extensive genetic correlations between diseases to achieve a positive knowledge transfer among diseases for PRS estimation.

As a limitation, we didn't perform any optimal task grouping. Leveraging optimal task grouping could improve the MTL model prediction performances.

This paper is currently under review.

5.4 Closing remarks and Future works

Overall, we achieved greater PRS estimation for many diseases. We proposed a new workflow to achieve optimal results for PRS on breast cancer leveraging deep neural network predicting power. Then, we developed LINA to enable enhanced interpretability for non-linear GWAS. Finally, we leveraged pleiotropy and common etiology with multi-LINA, a multi-task learning architecture inspired by LINA, to achieve higher

performance and uncover pleiotropic variants.

In the future, several possible research paths can be explored. The performance of our model on specific ancestry groups can be explored. Preprocessing steps utility, such as HWE and Linkage disequilibrium criteria relaxations, can be investigated to further validate deep neural networks' outperformance of the statistical models. Using more accurate peer-reviewed p-values for SNP filtering could also lead to better PRS prediction by deep neural networks and, thus, more accurate interpretations. Efficient Task grouping can also be researched to increase task-wise prediction performance. Finally, the development of methods to quantify uncertainty for model predictions and interpretation should be designed to enhance our proposed important SNP recovery method.

Reference List

, ????: pandas-plink/install.rst at main · limix/pandas-plink. URL https://github.com/limix/pandas-plink.

Abravaya, K., J. Huff, R. Marshall, B. Merchant, C. Mullen, G. Schneider, and J. Robinson, 2003: Molecular beacons as diagnostic tools: technology and applications.

Adeel, A., M. Gogate, and A. Hussain, 2020: Contextual deep learning-based audio-visual switching for speech enhancement in real-world environments. *Information Fusion*, **59**, 163–170.

Adewuyi, E. O., E. K. O'Brien, D. R. Nyholt, T. Porter, and S. M. Laws, 2022: A large-scale genome-wide cross-trait analysis reveals shared genetic architecture between Alzheimer's disease and gastrointestinal tract disorders. *Communications Biology*, **5 (1)**, 1–14, https://doi.org/10.1038/s42003-022-03607-2, URL https://www.nature.com/articles/s42003-022-03607-2, number: 1 Publisher: Nature Publishing Group.

Amos, C. I., and Coauthors, 2017: The oncoarray consortium: A network for understanding the genetic architecture of common cancersthe oncoarray and common cancer etiology. *Cancer epidemiology, biomarkers & prevention*, **26 (1)**, 126–135.

Angermueller, C., T. Pärnamaa, L. Parts, and O. Stegle, 2016: Deep learning for computational biology. *Molecular systems biology*, **12 (7)**, 878.

Badré, A., and C. Pan, 2022: Lina: A linearizing neural network architecture for accurate first-order and second-order interpretations. *IEEE Access*, **10**, 36 166–36 176.

Badré, A., L. Zhang, W. Muchero, J. C. Reynolds, and C. Pan, 2021: Deep neural network improves the estimation of polygenic risk scores for breast cancer. *Journal of Human Genetics*, **66 (4)**, 359–369.

Baltres, A., Z. Al Masry, R. Zemouri, S. Valmary-Degano, L. Arnould, N. Zerhouni, and C. Devalland, 2020: Prediction of oncotype dx recurrence score using deep multilayer perceptrons in estrogen receptor-positive, her2-negative breast cancer. *Breast Cancer*, **27**, 1007–1016.

Bellot, P., G. de Los Campos, and M. Pérez-Enciso, 2018: Can deep learning improve genomic prediction of complex human traits? *Genetics*, **210 (3)**, 809–819.

Bengio, Y., and Coauthors, 2009: Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, **2 (1)**, 1–127.

Bermeitinger., B., T. Hrycej., and S. Handschuh., 2019: Representational capacity of deep neural networks: A computing study. *Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR,*, SciTePress, INSTICC, 532-538, https://doi.org/10.5220/0008364305320538.

Bingel, J., and A. Søgaard, 2017: Identifying beneficial task relations for multi-task learning in deep neural networks. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, Association for Computational Linguistics, Valencia, Spain, 164–169, URL https://aclanthology.org/E17-2026.

Bishop, D. T., and Coauthors, 2009: Genome-wide association study identifies three loci associated with melanoma risk. *Nature Genetics*, **41 (8)**, 920–925, https://doi.org/10.1038/ng.411, URL https://www.nature.com/articles/ng.411, number: 8 Publisher: Nature Publishing Group.

Bonferroni, C. E., 1935: Il calcolo delle assicurazioni su gruppi di teste. *Studi in onore del professore salvatore ortu carboni*, 13–60.

Bracher-Smith, M., E. Rees, G. Menzies, J. T. R. Walters, M. C. O'Donovan, M. J. Owen, G. Kirov, and V. Escott-Price, 2022: Machine learning for prediction of schizophrenia using genetic and demographic factors in the UK biobank. *Schizophrenia Research*, **246**, 156–164, https://doi.org/10.1016/j.schres.2022.06.006, URL https://www.sciencedirect.com/science/article/pii/S0920996422002407.

Brandes, N., N. Linial, and M. Linial, 2021: Genetic association studies of alterations in protein function expose recessive effects on cancer predisposition. *Scientific Reports*, **11 (1)**, 14 901, https://doi.org/10.1038/s41598-021-94252-y, URL https://www.nature.com/articles/s41598-021-94252-y, number: 1 Publisher: Nature Publishing Group.

Bycroft, C., and Coauthors, 2018: The uk biobank resource with deep phenotyping and genomic data. *Nature*, **562 (7726)**, 203–209.

Byun, J., and Coauthors, 2021: The Shared Genetic Architectures Between Lung Cancer and Multiple Polygenic Phenotypes in Genome-Wide Association Studies. *Cancer Epidemiology, Biomarkers & Prevention*, **30 (6)**, 1156–1164, https://doi.org/10.1158/1055-9965.EPI-20-1635, URL https://doi.org/10.1158/1055-9965.EPI-20-1635.

Caruana, R., 1998: *Multitask learning*. Springer.

Cesaratto, L., and Coauthors, 2016: Bnc2 is a putative tumor suppressor gene in high-grade serous ovarian carcinoma and impacts cell survival after oxidative stress. *Cell death & disease*, **7 (9)**, e2374–e2374.

Chan, C. H. T., and Coauthors, 2018: Evaluation of three polygenic risk score models for the prediction of breast cancer risk in singapore chinese. *Oncotarget*, **9 (16)**, 12 796.

Chang, C. C., C. C. Chow, L. C. Tellier, S. Vattikuti, S. M. Purcell, and J. J. Lee, 2015: Second-generation plink: rising to the challenge of larger and richer datasets. *Gigascience*, **4 (1)**, s13 742–015.

Chen, J., L. Song, M. Wainwright, and M. Jordan, 2018: Learning to explain: An information-theoretic perspective on model interpretation. *International Conference on Machine Learning*, PMLR, 883–892.

Clark, S. A., B. P. Kinghorn, J. M. Hickey, and J. H. van der Werf, 2013: The effect of genomic information on optimal contribution selection in livestock breeding programs. *Genetics selection evolution*, **45**, 1–8.

Collobert, R., and J. Weston, 2008: A unified architecture for natural language processing: Deep neural networks with multitask learning. *Proceedings of the 25th international conference on Machine learning*, 160–167.

Consortium, . G. P., and Coauthors, 2015: A global reference for human genetic variation. *Nature*, **526 (7571)**, 68.

Cordell, H. J., 2002: Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. *Human molecular genetics*, **11 (20)**, 2463–2468.

Cudic, M., H. Baweja, T. Parhar, and S. T. Nuske, 2018: Prediction of sorghum bicolor genotype from in-situ images using autoencoder-identified snps. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, 23–31.

Cui, T., P. Marttinen, and S. Kaski, 2019: Learning global pairwise interactions with bayesian neural networks. *arXiv preprint arXiv:1901.08361*.

Dandl, S., C. Molnar, M. Binder, and B. Bischl, 2020: Multi-objective counterfactual explanations. *Parallel Problem Solving from Nature–PPSN XVI: 16th International Conference, PPSN 2020, Leiden, The Netherlands, September 5-9, 2020, Proceedings, Part I*, Springer, 448–469.

Dayem Ullah, A. Z., J. Oscanoa, J. Wang, A. Nagano, N. R. Lemoine, and C. Chelala, 2018: Snpnexus: assessing the functional relevance of genetic variation to facilitate the promise of precision medicine. *Nucleic acids research*, **46 (W1)**, W109–W113.

De, R., W. S. Bush, and J. H. Moore, 2014: Bioinformatics challenges in genome-wide association studies (gwas). *Clinical Bioinformatics*, 63–81.

DeBoever, C., Y. Tanigawa, M. Aguirre, G. McInnes, A. Lavertu, and M. A. Rivas, 2020: Assessing Digital Phenotyping to Enhance Genetic Studies of Human Diseases. *The American Journal of Human Genetics*, **106 (5)**, 611–622, https://doi.org/10.1016/j.ajhg.2020.03.007, URL https://www.cell.com/ajhg/abstract/S0002-9297(20)30083-5, publisher: Elsevier.

Deng, L., G. Hinton, and B. Kingsbury, 2013: New types of deep neural network learning for speech recognition and related applications: An overview. *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, 8599–8603.

Do, D. T., and N. Q. K. Le, 2020: Using extreme gradient boosting to identify origin of replication in saccharomyces cerevisiae via hybrid features. *Genomics*, **112 (3)**, 2445–2451.

Dudbridge, F., 2013: Power and predictive accuracy of polygenic risk scores. *PLoS genetics*, **9 (3)**, e1003 348.

El Naqa, I., and M. J. Murphy, 2015: *What is machine learning?* Springer.

Elgart, M., and Coauthors, 2022: Non-linear machine learning models incorporating SNPs and PRS improve polygenic prediction in diverse human populations. *Communications Biology*, **5 (1)**, 1–12, https://doi.org/10.1038/s42003-022-03812-z, URL https://www.nature.com/articles/s42003-022-03812-z, number: 1 Publisher: Nature Publishing Group.

Fergus, P., C. C. Montanez, B. Abdulaimma, P. Lisboa, C. Chalmers, and B. Pineles, 2018: Utilizing deep learning and genome wide association studies for epistatic-driven preterm birth classification in african-american women. *IEEE/ACM transactions on computational biology and bioinformatics*, **17 (2)**, 668–678.

Fernald, K., and M. Kurokawa, 2013: Evading apoptosis in cancer. *Trends in cell biology*, **23 (12)**, 620–633.

Fifty, C., E. Amid, Z. Zhao, T. Yu, R. Anil, and C. Finn, 2021: Efficiently Identifying Task Groupings for Multi-Task Learning. *Advances in Neural Information Processing Systems*, Curran Associates, Inc., Vol. 34, 27 503–27 516, URL https://proceedings.neurips.cc/paper/2021/hash/e77910ebb93b511588557806310f78f1-Abstract.html.

Friedman, J. H., 2001: Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189–1232.

Fukushima, K., 1975: Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, **20 (3-4)**, 121–136.

Gao, M., Y. Quan, X.-H. Zhou, and H.-Y. Zhang, 2019: Phewas-based systems genetics methods for anti-breast cancer drug discovery. *Genes*, **10 (2)**, 154.

Ge, T., C.-Y. Chen, Y. Ni, Y.-C. A. Feng, and J. W. Smoller, 2019: Polygenic prediction via bayesian regression and continuous shrinkage priors. *Nature communications*, **10 (1)**, 1776.

Ge, T., and Coauthors, 2022: Development and validation of a trans-ancestry polygenic risk score for type 2 diabetes in diverse populations. *Genome Medicine*, **14 (1)**, 70, https://doi.org/10.1186/s13073-022-01074-2, URL https://doi.org/10.1186/s13073-022-01074-2.

Gibbs, D., and Coauthors, 2017: Functional melanoma-risk variant IRF4 rs12203592 associated with Breslow thickness: a pooled international study of primary melanomas. *British Journal of Dermatology*, **177 (5)**, e180–e182, https://doi.org/10.1111/bjd.15784, URL https://doi.org/10.1111/bjd.15784.

Girshick, R., 2015: Fast r-cnn. *Proceedings of the IEEE international conference on computer vision*, 1440–1448.

Gola, D., J. Erdmann, B. Müller-Myhsok, H. Schunkert, and I. R. König, 2020: Polygenic risk scores outperform machine learning methods in predicting coronary artery disease status. *Genetic Epidemiology*, **44 (2)**, 125–138, https://doi.org/10.1002/gepi.22279, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/gepi.22279, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/gepi.22279.

Gonçalves, E., and Coauthors, 2020: Drug mechanism-of-action discovery through the integration of pharmacological and crispr screens. *Molecular Systems Biology*, **16 (7)**, e9405.

Hastie, T., S. Rosset, J. Zhu, and H. Zou, 2009: Multi-class adaboost. *Statistics and its Interface*, **2 (3)**, 349–360.

Henderson, C. R., 1975: Best linear unbiased estimation and prediction under a selection model. *Biometrics*, 423–447.

Ho, D. S. W., W. Schierding, M. Wake, R. Saffery, and J. O'Sullivan, 2019: Machine learning snp based prediction for precision medicine. *Frontiers in genetics*, **10**, 267.

Ho Thanh Lam, L., N. H. Le, L. Van Tuan, H. Tran Ban, T. Nguyen Khanh Hung, N. T. K. Nguyen, L. Huu Dang, and N. Q. K. Le, 2020: Machine learning model for identifying antioxidant proteins using features calculated from primary sequences. *Biology*, **9 (10)**, 325.

Hsieh, Y.-C., and Coauthors, 2017: A polygenic risk score for breast cancer risk in a taiwanese population. *Breast cancer research and treatment*, **163**, 131–138.

Ioffe, S., and C. Szegedy, 2015: Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv*, **abs/1502.03167**.

Jobs, M., 2001: Robust and accurate single nucleotide polymorphism genotyping by dynamic allele specific hybridization (dash). *SNP 2000: Third International Meeting on Single Nucleotide polymorphism and Complex Genome Analysis, Taos, New Mexico, USA, 2001*.

Joshi, A., S. Karimi, R. Sparks, C. Paris, and C. R. MacIntyre, 2019: Does Multi-Task Learning Always Help?: An Evaluation on Health Informatics. *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, Australasian Language Technology Association, Sydney, Australia, 151–158, URL https://aclanthology.org/U19-1020.

Khera, A. V., and Coauthors, 2018: Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations. *Nature genetics*, **50 (9)**, 1219–1224.

Kingma, D. P., and J. Ba, 2014a: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., and J. Ba, 2014b: Adam: A method for stochastic optimization. *CoRR*, **abs/1412.6980**.

Kolch, W., M. Halasz, M. Granovskaya, and B. N. Kholodenko, 2015: The dynamic control of signal transduction networks in cancer cells. *Nature Reviews Cancer*, **15 (9)**, 515–527.

Koppe, G., A. Meyer-Lindenberg, and D. Durstewitz, 2021: Deep learning for small and big data in psychiatry. *Neuropsychopharmacology*, **46 (1)**, 176–190.

Landi, M. T., and Coauthors, 2020: Genome-wide association meta-analyses combining multiple risk phenotypes provide insights into the genetic architecture of cutaneous melanoma susceptibility. *Nature Genetics*, **52 (5)**, 494–504, https://doi.org/10.1038/s41588-020-0611-8, URL https://www.nature.com/articles/s41588-020-0611-8, number: 5 Publisher: Nature Publishing Group.

Lawson, D. J., and Coauthors, 2020: Is population structure in the genetic biobank era irrelevant, a challenge, or an opportunity? *Human Genetics*, **139**, 23–41.

LeBlanc, M., and C. Kooperberg, 2010: Boosting predictions of treatment success. *Proceedings of the National Academy of Sciences*, **107 (31)**, 13 559–13 560.

LeCun, Y., L. Bottou, Y. Bengio, and P. Haffner, 1998: Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, **86 (11)**, 2278–2324.

Lee, J.-Y., and Coauthors, 2009: Candidate gene approach evaluates association between innate immunity genes and breast cancer risk in korean women. *Carcinogenesis*, **30 (9)**, 1528–1531.

Li, X., Z. Zou, J. Tang, Y. Zheng, Y. Liu, Y. Luo, Q. Liu, and Y. Wang, 2019: Nos1 upregulates abcg2 expression contributing to ddp chemoresistance in ovarian cancer cells. *Oncology letters*, **17 (2)**, 1595–1602.

Liu, P., X. Qiu, and X. Huang, 2016: Recurrent neural network for text classification with multi-task learning. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, AAAI Press, New York, New York, USA, 2873–2879, IJCAI'16.

Lu, L., Y. Shin, Y. Su, and G. E. Karniadakis, 2019: Dying relu and initialization: Theory and numerical examples. *arXiv preprint arXiv:1903.06733*.

Lundberg, S. M., and S.-I. Lee, 2017: A unified approach to interpreting model predictions. *Advances in neural information processing systems*, **30**.

Lutz, M. W., D. Sprague, J. Barrera, and O. Chiba-Falek, 2020: Shared genetic etiology underlying Alzheimer's disease and major depressive disorder. *Translational Psychiatry*, **10 (1)**, 1–14, https://doi.org/10.1038/s41398-020-0769-y, URL https://www.nature.com/articles/s41398-020-0769-y, number: 1 Publisher: Nature Publishing Group.

Maas, A. L., A. Y. Hannun, A. Y. Ng, and Coauthors, 2013: Rectifier nonlinearities improve neural network acoustic models. *Proc. icml*, Atlanta, Georgia, USA, Vol. 30, 3.

Maier, R., and Coauthors, 2015: Joint analysis of psychiatric disorders increases accuracy of risk prediction for schizophrenia, bipolar disorder, and major depressive disorder. *The American Journal of Human Genetics*, **96 (2)**, 283–294.

Mailman, M. D., and Coauthors, 2007: The ncbi dbgap database of genotypes and phenotypes. *Nature genetics*, **39 (10)**, 1181–1186.

Mao, Q., and J. D. Unadkat, 2015: Role of the breast cancer resistance protein (bcrp/abcg2) in drug transport—an update. *The AAPS journal*, **17**, 65–82.

Marchini, J., L. R. Cardon, M. S. Phillips, and P. Donnelly, 2004: The effects of human population structure on large genetic association studies. *Nature genetics*, **36 (5)**, 512–517.

Mavaddat, N., and Coauthors, 2019: Polygenic risk scores for prediction of breast cancer and breast cancer subtypes. *The American Journal of Human Genetics*, **104 (1)**, 21–34.

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

Michailidou, K., and Coauthors, 2017: Association analysis identifies 65 new breast cancer risk loci. *Nature*, **551 (7678)**, 92–94.

Miller, T., 2019: Explanation in artificial intelligence: Insights from the social sciences. *Artificial intelligence*, **267**, 1–38.

Molnar, C., 2020: *Interpretable machine learning*. Lulu. com.

Márquez-Luna, C., S. Gazal, P.-R. Loh, S. S. Kim, N. Furlotte, A. Auton, and A. L. Price, 2021: Incorporating functional priors improves polygenic prediction accuracy in UK Biobank and 23andMe data sets. *Nature Communications*, **12 (1)**, 6052, https://doi.org/10.1038/s41467-021-25171-9, URL https://www.nature.com/articles/s41467-021-25171-9, number: 1 Publisher: Nature Publishing Group.

Nelson, H. D., K. Tyne, A. Naik, C. Bougatsos, B. K. Chan, and L. Humphrey, 2009: Screening for breast cancer: an update for the us preventive services task force. *Annals of internal medicine*, **151 (10)**, 727–737.

NIH, 2012: Cancer of the breast (female) - cancer stat facts. URL https://seer.cancer.gov/statfacts/html/breast.html.

Novembre, J., and Coauthors, 2008: Genes mirror geography within europe. *Nature*, **456 (7218)**, 98–101.

Oeffinger, K. C., and Coauthors, 2015: Breast cancer screening for women at average risk: 2015 guideline update from the american cancer society. *Jama*, **314 (15)**, 1599–1614.

O'Connor, M. J., 2015: Targeting the dna damage response in cancer. *Molecular cell*, **60 (4)**, 547–560.

Pace, R. K., and R. Barry, 1997: Sparse spatial autoregressions. *Statistics & Probability Letters*, **33 (3)**, 291–297.

Phillips, P. C., 2008: Epistasis—the essential role of gene interactions in the structure and evolution of genetic systems. *Nature Reviews Genetics*, **9 (11)**, 855–867.

Privé, F., J. Arbel, and B. J. Vilhjálmsson, 2020: LDpred2: better, faster, stronger. *Bioinformatics*, **36 (22-23)**, 5424–5431, https://doi.org/10.1093/bioinformatics/btaa1029, URL https://doi.org/10.1093/bioinformatics/btaa1029.

Purcell, S., and Coauthors, 2007: Plink: a tool set for whole-genome association and population-based linkage analyses. *The American journal of human genetics*, **81 (3)**, 559–575.

Purcell Shaun, S. J. V. P. O. M. C. . S. P. F. . S. P. . . . ., Wray Naomi, and Coauthors, 2009: Common polygenic variation contributes to risk of schizophrenia and bipolar disorder. *Nature*, **460 (7256)**, 748–752.

Ransohoff, K. J., and Coauthors, 2017: Two-stage genome-wide association study identifies a novel susceptibility locus associated with melanoma. *Oncotarget*, **8 (11)**, 17 586–17 592, https://doi.org/10.18632/oncotarget.15230, URL https://www.oncotarget.com/article/15230/text/, publisher: Impact Journals.

Rashkin, S. R., and Coauthors, 2020: Pan-cancer study detects genetic risk variants and shared genetic basis in two large cohorts. *Nature Communications*, **11 (1)**, 4423, https://doi.org/10.1038/s41467-020-18246-6, URL https://www.nature.com/articles/s41467-020-18246-6, number: 1 Publisher: Nature Publishing Group.

Ribeiro, M. T., S. Singh, and C. Guestrin, 2016: " why should i trust you?" explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135–1144.

Rivandi, M., J. W. Martens, and A. Hollestelle, 2018: Elucidating the underlying functional mechanisms of breast cancer susceptibility through post-gwas analyses. *Frontiers in genetics*, **9**, 280.

Romualdo Cardoso, S., A. Gillespie, S. Haider, and O. Fletcher, 2022: Functional annotation of breast cancer risk loci: current progress and future directions. *British Journal of Cancer*, **126 (7)**, 981–993.

Rosenblatt, F., 1958: The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, **65 (6)**, 386.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams, 1986: Learning representations by back-propagating errors. *nature*, **323 (6088)**, 533–536.

Schmidhuber, J., 2015: Deep learning in neural networks: An overview. *Neural networks*, **61**, 85–117.

Scott, R. A., and Coauthors, 2017: An expanded genome-wide association study of type 2 diabetes in europeans. *Diabetes*, **66 (11)**, 2888–2902.

Shaker, O. G., and M. A. Senousy, 2019: Association of snp-snp interactions between rankl, opg, chi3l1, and vdr genes with breast cancer risk in egyptian women. *Clinical Breast Cancer*, **19 (1)**, e220–e238.

Shrikumar, A., P. Greenside, and A. Kundaje, 2017: Learning important features through propagating activation differences. *International conference on machine learning*, PMLR, 3145–3153.

Simonyan, K., A. Vedaldi, and A. Zisserman, 2014: Deep inside convolutional networks: Visualising image classification models and saliency maps. *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Workshop Track Proceedings*, Y. Bengio, and Y. LeCun, Eds., URL http://arxiv.org/abs/1312.6034.

Sorokina, D., R. Caruana, and M. Riedewald, 2007: Additive groves of regression trees. *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings 18*, Springer, 323–334.

Speed, D., and D. J. Balding, 2014: Multiblup: improved snp-based prediction for complex traits. *Genome research*, **24 (9)**, 1550–1557.

Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, 2014: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, **15**, 1929–1958.

Standley, T., A. R. Zamir, D. Chen, L. Guibas, J. Malik, and S. Savarese, 2019: Which Tasks Should Be Learned Together in Multi-task Learning? URL https://openreview.net/forum?id=HJlTpCEKvS.

Steemers, F. J., and K. L. Gunderson1, 2, 2005: Illumina, inc.

Sutton, M., P.-E. Sugier, T. Truong, and B. Liquet, 2022: Leveraging pleiotropic association using sparse group variable selection in genomics data. *BMC Medical Research Methodology*, **22 (1)**, 9, https://doi.org/10.1186/s12874-021-01491-8, URL https://doi.org/10.1186/s12874-021-01491-8.

Thissen, J. B., and Coauthors, 2019: Axiom microbiome array, the next generation microarray for high-throughput pathogen and microbiome analysis. *PLoS One*, **14 (2)**, e0212 045.

Thomas, M., and Coauthors, 2020: Genome-wide Modeling of Polygenic Risk Score in Colorectal Cancer Risk. *The American Journal of Human Genetics*, **107 (3)**, 432–444, https://doi.org/10.1016/j.ajhg.2020.07.006, URL https://www.sciencedirect.com/science/article/pii/S0002929720302366.

Tian, H., S.-C. Chen, and M.-L. Shyu, 2020: Evolutionary programming based deep learning feature selection and network construction for visual data classification. *Information systems frontiers*, **22**, 1053–1066.

Tinholt, M., and Coauthors, 2014: Increased coagulation activity and genetic polymorphisms in the f5, f10 and epcr genes are associated with breast cancer: a case-control study. *Bmc Cancer*, **14**, 1–11.

Tsang, M., D. Cheng, and Y. Liu, 2017: Detecting statistical interactions from neural network weights. *International Conference on Machine Learning*.

Tsuboi, M., and Coauthors, 2019: Prognostic significance of gad1 overexpression in patients with resected lung adenocarcinoma. *Cancer medicine*, **8 (9)**, 4189–4199.

Uffelmann, E., and Coauthors, 2021: Genome-wide association studies. *Nature Reviews Methods Primers*, **1 (1)**, 59.

van de Haar, J., S. Canisius, K. Y. Michael, E. E. Voest, L. F. Wessels, and T. Ideker, 2019: Identifying epistasis in cancer genomes: a delicate affair. *Cell*, **177 (6)**, 1375–1383.

Velasco-Ruiz, A., and Coauthors, 2021: Polrmt as a novel susceptibility gene for cardiotoxicity in epirubicin treatment of breast cancer patients. *Pharmaceutics*, **13 (11)**, 1942.

Vilhjálmsson, B. J., and Coauthors, 2015: Modeling linkage disequilibrium increases accuracy of polygenic risk scores. *The american journal of human genetics*, **97 (4)**, 576–592.

Wang, L., J. Ingle, and R. Weinshilboum, 2018: Pharmacogenomic discovery to function and mechanism: breast cancer as a case study. *Clinical Pharmacology & Therapeutics*, **103 (2)**, 243–252.

Watanabe, K., and Coauthors, 2019: A global overview of pleiotropy and genetic architecture in complex traits. *Nature Genetics*, **51 (9)**, 1339–1348, https://doi.org/10.1038/s41588-019-0481-0, URL https://www.nature.com/articles/s41588-019-0481-0, number: 9 Publisher: Nature Publishing Group.

Wei, Z., and Coauthors, 2009: From disease association to risk assessment: an optimistic view from genome-wide association studies on type 1 diabetes. *PLoS genetics*, **5 (10)**, e1000 678.

Wen, W., and Coauthors, 2016: Prediction of breast cancer risk based on common genetic variants in women of east asian ancestry. *Breast Cancer Research*, **18 (1)**, 1–8.

Whittaker, A. J., I. Royzman, and T. L. Orr-Weaver, 2000a: Drosophila double parked: a conserved, essential replication protein that colocalizes with the origin recognition complex and links dna replication with mitosis and the down-regulation of s phase transcripts. *Genes & development*, **14 (14)**, 1765–1776.

Whittaker, A. J., I. Royzman, and T. L. Orr-Weaver, 2000b: Drosophila Double parked: a conserved, essential replication protein that colocalizes with the origin recognition complex and links DNA replication with mitosis and the down-regulation of S phase transcripts. *Genes & Development*, **14 (14)**, 1765–1776, https://doi.org/10.1101/gad.14.14.1765, URL http://genesdev.cshlp.org/content/14/14/1765, company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Institution: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Publisher: Cold Spring Harbor Lab.

Woo, A., S. W. Lee, H. Y. Koh, M. A. Kim, M. Y. Han, and D. K. Yon, 2021: Incidence of cancer after asthma development: 2 independent population-based cohort studies.

Journal of Allergy and Clinical Immunology, **147 (1)**, 135–143, https://doi.org/10.1016/j.jaci.2020.04.041, URL https://www.jacionline.org/article/S0091-6749(20)30643-6/fulltext, publisher: Elsevier.

Wu, L., and Coauthors, 2016: A genome-wide association study identifies wt1 variant with better response to 5-fluorouracil, pirarubicin and cyclophosphamide neoadjuvant chemotherapy in breast cancer patients. *Oncotarget*, **7 (4)**, 5042.

Wu, X., and Coauthors, 2022: Investigating the shared genetic architecture of uterine leiomyoma and breast cancer: A genome-wide cross-trait analysis. *The American Journal of Human Genetics*, **109 (7)**, 1272–1285, https://doi.org/10.1016/j.ajhg.2022.05.015, URL https://www.cell.com/ajhg/abstract/S0002-9297(22)00253-1, publisher: Elsevier.

Xu, B., N. Wang, T. Chen, and M. Li, 2015: Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*.

Ye, Y., X. Chen, J. Han, W. Jiang, P. Natarajan, and H. Zhao, 2021: Interactions Between Enhanced Polygenic Risk Scores and Lifestyle for Cardiovascular Disease, Diabetes, and Lipid Levels. *Circulation: Genomic and Precision Medicine*, **14 (1)**, e003 128, https://doi.org/10.1161/CIRCGEN.120.003128, URL https://www.ahajournals.org/doi/full/10.1161/CIRCGEN.120.003128, publisher: American Heart Association.

Yin, B., M. Balvert, R. A. van der Spek, B. E. Dutilh, S. Bohté, J. Veldink, and A. Schönhuth, 2019: Using the structure of genome data in the design of deep neural networks for predicting amyotrophic lateral sclerosis from genotype. *Bioinformatics*, **35 (14)**, i538–i547.

Young, T., D. Hazarika, S. Poria, and E. Cambria, 2018: Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, **13 (3)**, 55–75.

Zhang, Y., and Q. Yang, 2018: An overview of multi-task learning. *National Science Review*, **5 (1)**, 30–43.

Zhao, X., J. Li, Z. Liu, and S. Powers, 2021: Combinatorial crispr/cas9 screening reveals epistatic networks of interacting tumor suppressor genes and therapeutic targets in human breast cancer. *Cancer Research*.

Zhou, Y., Z.-S. Liang, Y. Jin, J. Ding, T. Huang, J. H. Moore, Z.-J. Zheng, and J. Huang, 2022: Shared Genetic Architecture and Causal Relationship Between Asthma and Cardiovascular Diseases: A Large-Scale Cross-Trait Analysis. *Frontiers in Genetics*, **12**, URL https://www.frontiersin.org/articles/10.3389/fgene.2021.775591.

Zhuang, Z., M. Yao, J. Y. Y. Wong, Z. Liu, and T. Huang, 2021: Shared genetic etiology and causality between body fat percentage and cardiovascular diseases: a large-scale genome-wide cross-trait analysis. *BMC Medicine*, **19 (1)**, 100, https://doi.org/10.1186/s12916-021-01972-z, URL https://doi.org/10.1186/s12916-021-01972-z.