Sparse Bayesian Learning for Predicting Phenotypes and Identifying Influential Markers

by

Maryam Ayat

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Doctor of Philosophy

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
December 2018

Thesis advisor
Michael Domaratzki

Author
Maryam Ayat

Sparse Bayesian Learning for Predicting Phenotypes and Identifying Influential Markers

Abstract

In bioinformatics, Genomic Selection (GS) and Genome-Wide Association Studies (GWASs) are two related problems that can be applied to the plant breeding industry. GS is a method to predict phenotypes (i.e., traits) such as yield and disease resistance in crops from high-density markers positioned throughout the genome of the varieties. By contrast, a GWAS involves identifying markers or genes that underlie the phenotypes of importance in breeding. The need to accelerate the development of improved varieties, and challenges such as discovering all sorts of genetic factors related to a trait, increasingly persuade researchers to apply state-of-the-art machine learning methods to GS and GWASs.

The aim of this study is to employ sparse Bayesian learning as a technique for GS and GWAS. The sparse Bayesian learning uses Bayesian inference to obtain sparse solutions in regression or classification problems. This learning method is also called the Relevance Vector Machine (RVM), as it can be viewed as a kernel-based model of identical form to the renowned Support Vector Machine (SVM) method.

The RVM has some advantages that the SVM lacks, such as having probabilistic outputs, providing a much sparser model, and the ability to work with arbitrary kernel functions. However, despite the advantages, there is not enough research on the applicability of the RVM.

In this thesis, we define and explore two different forms of the sparse Bayesian learning for predicting phenotypes and identifying the most influential markers of a trait, respectively. Particularly, we introduce a new framework based on sparse Bayesian learning and ensemble technique for ranking influential markers of a trait. We apply our methods on three different datasets, one simulated dataset and two real-world datasets (yeast and flax), and analyze our results with respect to the existing related works, trait heritability, and the accuracies obtained from the use of different kernel functions including linear, Gaussian, and string kernels, if applicable. We find that the RVMs can not only be considered as good as other successful machine learning methods in phenotype prediction, but are also capable of identifying the most important markers from which biologists might gain insight.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to begin by thanking my advisor, Dr. Mike Domaratzki, for all he has done for me during my graduate program.

I would also like to thank the rest of my examining committee, Dr. Cory Butz, Dr. Elif Acar, and Dr. Olivier Tremblay-Savard, for their insightful comments and suggestions which improved this thesis.

Also, I am grateful that I have had Soheil, a great supporter, and Hossein, a unique source of hope and positivity, by my side. I would never have been able to complete my program without your enthusiastic encouragement. Thanks for everything.

This thesis is dedicated to my beloved mother, Mehraneh Moallem.

Chapter 1

Introduction

In this thesis, we address two related problems in bioinformatics: Genomic Selection and Genome-Wide Association Studies (GWASs). Genomic selection involves predicting phenotypes such as growth and fertility in livestock [34, 50], yield and drought resistance in crops [40], and disease risk in humans [1, 41], using genetic information of individuals (i.e., sequences of genome-wide molecular markers). The main aim of genomic selection is maximization of predictive power. Genomic selection is ideal for complex traits, which are controlled by many genes with different effects across the genome [72]. Genomic selection in plants or animals is mainly used in the breeding industry.

On the other hand, GWAS deals with causal interpretation of phenotypic variations in humans, plants, or animals. GWAS helps to understand the genetic architecture of complex traits. For example, GWAS in humans involves rapidly scanning

markers across the genomes of many people to find genetic variations associated with a particular disease. Once new genetic associations are identified, researchers can use the information to develop better strategies to detect, treat and prevent the disease [47]. GWAS, particularly in humans, has yielded the "missing heritability" problem [60], though it is also present in other organisms (e.g., [80, 100]). Missing heritability is the gap between known and predicted heritability. In other words, GWAS has identified many genetic loci for a wide range of traits, but these typically explain only a minority of the heritability of each trait, implying the existence of other undiscovered genetic factors. An example of this missing heritability is height in humans, and disease resistance in crops. However, devising new approaches and proper tools in GWAS may uncover the missing part [100].

There are a variety of computational models used in genomic selection and GWAS, though mostly statistical (e.g., Best Linear Unbiased Prediction, or BLUP [43]), traditionally. However, machine learning methods, such as random forests [16] and Support Vector Machines (SVMs) [79], have had an increasing interest to overcome challenges in these problems, such as identifying markers that influence a trait in complex interactions [36, 89].

In this thesis, we employ the sparse Bayesian learning method [92] for predicting phenotypes and identifying influential markers of a trait. The sparse Bayesian learning uses Bayesian inference to obtain sparse solutions for regression and classification tasks. This learning method is also called the Relevance Vector Machine

(RVM), as it can be viewed as a kernel-based model of identical form to the SVM. SVMs are one of the most theoretically well-motivated and practically most effective classification algorithms in modern machine learning, as expressed by Mohri et al. [64]. Although the prediction performance of the RVM practically competes with the SVM, it has some advantages that the SVM lacks, e.g, having probabilistic outputs, providing a much sparser model, and the ability to work with arbitrary kernel functions. However, there is not enough research about sparse Bayesian learning applicability, particularly in bioinformatics, despite its advantages.

In this work, we have two major contributions: (1) We employ RVMs with different kernels for predicting phenotypes via regression or classification. We also use ensemble RVM to improve prediction accuracy and rank the training samples (or variants). Ranking individuals allows us to extract the most informative samples, and consequently, to get more insights from data. To the best of our knowledge, there has not been any research on the application of RVM in genomic selection. Also we did not find any work which used ensemble RVM for ranking purposes. (2) We merge sparse Bayesian learning and ensemble technique for feature selection and ranking, i.e., identifying influential markers of a trait. This is a new approach, and such an architecture has not been used for feature selection previously. We apply our methods on three different datasets, one simulated dataset and two real-world datasets (yeast and flax), and analyze our results in respect of the existing related works, trait heritability, and the accuracies obtained from the use of different kernel

functions including linear, Gaussian, and string kernels.

We have organized the thesis material as follows. In Chapter 2, we describe the required machine learning background including sparse Bayesian learning in classification and regression. Next in Chapter 3, we represent our proposed methods, i.e., Kernel RVMs, Basis RVMs, and Ensemble RVMs. Then in Chapter 4, we illustrate the bioinformatics scope of this thesis, i.e., genomic selection and GWAS. We also review existing related work for prediction and feature selection in bioinformatics. Next in Chapter 5, we investigate how our RVM architectures perform on a simulated data set. Then in Chapters 6 and 7, we experiment the RVMs on two real world datasets, yeast and flax, respectively. Last, we present conclusions and future work in Chapter 8.

Chapter 2

Background

In this chapter, we describe the machine learning background required to understand the proposed methods and the related work, such as Machine Learning, Kernel Methods, Sparse Bayesian Learning and Relevance Vector Machine (RVM), and Evaluation Measures and Techniques.

2.1 Machine Learning

Machine learning [64], as a part of Artificial Intelligence, is a branch of computer science that gives computers the ability to learn from examples or past experiences, and detect patterns in data or make predictions on data. For example, an email spam filter uses machine learning to sort incoming mails into spams and non-spams. If a user frequently discards emails with a specific header, the spam filter will start

to categorize similar emails as unwanted or spams.

Machine learning algorithms are often categorized into two types: *supervised* and *unsupervised*. *Supervised learning* is the task of inferring a function from *labelled* training data. The training data is a set of training examples, or pairs, each consisting of an input object (which is typically represented as a vector) and a desired output value. The goal of the algorithm is to correctly determine the output value for new, unseen examples. Two major categories of supervised learning problems are classification and regression. A classification assigns a category or target value to each item (e.g., a spam filter assigns spam/non-spam categories to emails), while a regression predicts a real value for each item (e.g., predicting house prices based on features of residential houses sold in a duration). The performance of the resulting function should be measured on a test set which is separate from the training set and consists of unseen instances.

Contrary to supervised learning, we only have input data in *unsupervised learning*. In this case, the learning task is inferring a function to describe the hidden structure in *unlabelled* data. For instance, grouping customers by purchasing behaviour can be viewed as a clustering problem which is an unsupervised learning method.

In the machine learning context, the set of all possible examples or instances is referred to as the *input space*, and the set of attributes associated to an example is called the *feature vector*. In the email spam filter problem wherein instances are email messages, some relevant features may include the length of the message, the

name of the sender, and the presence of certain keywords in the header or body of the message. In fact, a feature vector is an n-dimensional vector of n numerical features that represents some example. The vector space associated with the feature vectors is called the *feature space*.

Machine learning techniques are increasingly being used to address problems in bioinformatics in which there is a large amount of data and noisy patterns, but no general theory for describing those data. Some of the widely used machine learning techniques in bioinformatics are Artificial Neural Networks [9], Support Vector Machines [11], Hidden Markov Models [105], Bayesian Networks [67], and Decision Trees [85].

2.2 Kernel Methods

Kernel methods, such as SVM, are a class of machine learning algorithms that depend on data only through dot products. The dot product of two vectors defines a similarity measure between the pair. In a kernel method, a *kernel function* computes a dot product in some possibly high-dimensional feature space. An advantage of this technique is the ability to generate non-linear decision boundaries.

For instance, consider the left hand side of Figure 2.1, and suppose our classification task is discriminating between crosses and circles in the input space. However, no line (or hyperplane in higher dimensional examples) can separate the two populations in that space. In fact, it is likely that in a real classification pattern recognition

problem, linear separation is not possible, as is shown in the left hand side of the figure. In such a case, one solution is to use a non-linear mapping of the points into some higher-dimensional feature space in which the samples are linearly separable. Then, we solve the problem (i.e., finding the optimal hyperplane) in the feature space, and consequently, we will be able to identify the corresponding non-linear decision boundary for the points in the input space. To do this procedure, a kernel method only requires a function $K : X \times X \longrightarrow \mathbb{R}$, which is called a kernel over the input space X. For any two input vectors $\mathbf{x}_i, \mathbf{x}_j \in X$, $K(\mathbf{x}_i, \mathbf{x}_j)$ is the dot product of vectors $\varphi(\mathbf{x}_i)$ and $\varphi(\mathbf{x}_j)$:

$$\forall \mathbf{x}_i, \mathbf{x}_j \in X, \quad K(\mathbf{x}_i, \mathbf{x}_j) = \langle \varphi(\mathbf{x}_i), \varphi(\mathbf{x}_j) \rangle \,, \tag{2.1}$$

for some mapping $\varphi : X \longrightarrow H$ to a feature space H. The kernel also can be written in matrix form over the data sample: $\mathbf{K} = [K(\mathbf{x}_i, \mathbf{x}_j)]_{ij}$, which is called the *kernel matrix* or *Gram matrix*.

In fact, the kernel method operates in a high-dimensional, implicit feature space without needing to know the mapping function φ, as computing the inner products between the images of all pairs of data points in the feature space suffices. However, the appropriate choice of a kernel is left to the user. This approach is called the kernel trick, and Figure 2.1 shows a graphical illustration for it in a binary classification example. As data points are not linearly separable in the input space, they are mapped from the input space to a feature space, using the mapping $\varphi : \mathbb{R}^2 \to \mathbb{R}^3, \varphi(\mathbf{x}) = \mathbf{z}$.

The mapping φ is defined implicitly via the kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle^2$. One possible mapping for this example can be $\varphi(x_1, x_2) = ({x_1}^2, {x_2}^2, \sqrt{2}x_1x_2) = (z_1, z_2, z_3)$. The images of data points in the feature space are linearly separable by a plane.

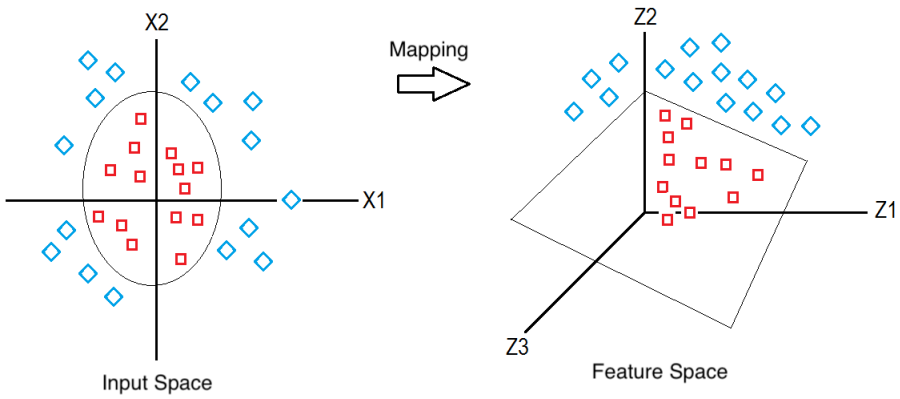


Figure 2.1: Mapping data points from an input space to a feature space.

The main advantage of using kernel functions is that computing the kernel is easy, but computing the feature vector corresponding to the kernel is mostly hard and costly, or even impossible. For example, the corresponding feature vector for a simple kernel such as Gaussian kernel (2.2) has an infinite number of dimensions, but computing the kernel itself is trivial.

To guarantee the convergence in a kernel method such as SVM, the kernel function K must satisfy Mercer's condition, which says the square kernel matrix should be Positive Definite Symmetric (PDS) [46], such as is the case for polynomial and Gaussian kernels. More precisely, satisfying Mercer's condition guarantees existence of an underlying mapping functions in (2.1). Otherwise, the kernel function is not

legitimate in SVM, and the classification problem cannot be solved. However, there is no such limitation in an RVM framework [92], as we will see later in this chapter.

2.2.1 Kernel Types

In this research, we use both sequence and non-sequence kernel functions. A non-sequence kernel refers to a kernel that can handle binary or numerical data types (e.g., gene expression data). Gaussian kernel and polynomial kernel are among non-sequence kernels: For any constant $\gamma > 0$, Gaussian kernel or Radial Basis Function (RBF) is the kernel $K : \mathbb{R}^N \to \mathbb{R}$:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^N, \quad K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2), \tag{2.2}$$

where $\|\mathbf{x}\|$ is the norm of the vector $\mathbf{x}$. Also, a polynomial kernel of degree d such as K is defined by:

$$\forall \mathbf{x}, \mathbf{x}' \in \mathbb{R}^N, \quad K(\mathbf{x}, \mathbf{x}') = (\mathbf{x} \cdot \mathbf{x}' + c)^d, \tag{2.3}$$

for a fixed constant $c \geq 0$.

In contrast to a non-sequence kernel, a sequence kernel operates on strings, or finite sequences of symbols. Intuitively speaking, we can say that the more similar the two strings $\mathbf{x}$ and $\mathbf{x}'$ are, the higher the value of a string kernel $K(\mathbf{x}, \mathbf{x}')$ will be. Sequence kernels have applications in biological sequence analysis, natural language

processing, document classification and other text processing areas [82].The n-gram kernel [59] is an example of a sequence kernel. The n-gram kernel of the two strings $\mathbf{x}$ and $\mathbf{x}'$ counts how often each contiguous string of length n is contained in the strings:

$$K_n(\mathbf{x}, \mathbf{x}') = \sum_{u \in A^n} \psi_u(\mathbf{x}) \psi_u(\mathbf{x}'), \tag{2.4}$$

where $\psi_u(\mathbf{x})$ denotes the number of occurrences of the subsequence u in the string $\mathbf{x}$, and A^n is the set of all possible subsequence of length n, given the alphabet A. For instance, suppose we are given two DNA sequences with the alphabet $A = \{\text{A,C,G,T}\}$: $\mathbf{x} = \text{AACCT}$ and $\mathbf{x}' = \text{GACAC}$. The bi-gram (2-gram) subsequences in $\mathbf{x}$ and $\mathbf{x}'$ are $\{\text{AA,AC,CC,CT}\}$ and $\{\text{GA,AC,CA}\}$, respectively. Therefore, $K_2(\mathbf{x}, \mathbf{x}') = 1 \times 2 = 2$, as only one subsequence AC is common in both sequences, which it has been repeated once in $\mathbf{x}$ and twice in $\mathbf{x}'$. Higher kernel values mean two sequences are more similar. Other than n-gram kernels, there are also other sequence kernels common in bioinformatics, such as mismatch, gappy, substitution, and homology kernels [28, 55, 59].

Most sequence kernels used in applications such as computational biology and natural language processing are considered rational kernels [23]. Rational kernels [21, 22], which are based on finite-state transducers [73], present an efficient general algorithm for manipulating variable-length sequence data.

2.3 Sparse Bayesian Learning

The sparse Bayesian modelling [92, 93] is an approach for learning the prediction function $y(\mathbf{x}; \mathbf{w})$, which is expressed as a linear combination of basis functions:

$$y(\mathbf{x}; \mathbf{w}) = \sum_{m=1}^{M} w_m \phi_m(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}), \tag{2.5}$$

where $\boldsymbol{\phi}(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_M(\mathbf{x}))^T$ are basis functions, generally non-linear, and $\mathbf{w} = (w_1, ..., w_M)^T$ are the adjustable parameters, called weights. Given a dataset of input-target training pairs $\{(\mathbf{x}_i, t_i)\}_{i=1}^N$, the objective of the sparse Bayesian method is to estimate the target function $y(\mathbf{x}; \mathbf{w})$, while retaining as few basis functions as possible. The sparse Bayesian algorithm often generates exceedingly sparse solutions (i.e., few non-zero parameters w_i).

In a particular specialization of (2.5), such as the one that SVM uses, $M = N$ and the basis functions take the form of kernel functions, one for each data point $\mathbf{x}_m$ in the training set, so that $\phi_m(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}_m)$, where $K(.,.)$ is the kernel function. This exemplification of the sparse Bayesian modelling is called the Relevance Vector Machine (RVM). Tipping [91] introduced the RVM method as an alternative to the SVM method of Vapnik [94]. However unlike the SVM, we can use arbitrary basis sets in the RVM [92], as the sparse Bayesian framework permits.

Assuming that the basis functions have the form of kernel functions, we illustrate the sparse Bayesian algorithms for regression and classification in the following

section (in order to facilitate direct comparisons with the SVM). Corresponding algorithms for arbitrary basis functions can be easily induced from them.

2.3.1 Relevance Vector Regression

We follow the framework developed by Tipping [92]. In the regression framework, the targets $\mathbf{t} = (t_1, ..., t_N)^T$ are real-valued labels. Each target t_i is representative of the true model y_i, but with the addition of noise ϵ_i: $t_i = y(\mathbf{x}_i) + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2)$. This means: $p(t_i \mid \mathbf{x}_i, \mathbf{w}, \sigma^2) = N(y(\mathbf{x}_i), \sigma^2)$, or:

$$p\left(\mathbf{t} \mid \mathbf{w}, \sigma^2\right) = (2\pi\sigma^2)^{-N/2} \exp\left\{-\frac{1}{2\sigma^2}\|\mathbf{t} - \mathbf{\Phi}\mathbf{w}\|^2\right\}, \tag{2.6}$$

where $\mathbf{w} = (w_1, ..., w_N)^T$, and the data is hidden in the design matrix (kernel matrix) $\mathbf{\Phi} = [\boldsymbol{\phi}(\mathbf{x}_1), ..., \boldsymbol{\phi}(\mathbf{x}_N)]^T$, wherein $\boldsymbol{\phi}(\mathbf{x}_i) = [K(\mathbf{x}_i, \mathbf{x}_1), ..., K(\mathbf{x}_i, \mathbf{x}_N)]^T$. For simplicity, we omit the implicit conditioning on the set of input vectors $\{\mathbf{x}_i\}$ in (2.6) and subsequent expression.

We infer weights using a fully probabilistic framework. Specifically, we define a Gaussian prior distribution with zero mean and ${\alpha_i}^{-1}$ variance over each w_i: $p(w_i \mid \alpha_i) = N(0, {\alpha_i}^{-1})$, or:

$$p(\mathbf{w} \mid \boldsymbol{\alpha}) = \prod_{i=1}^{N} N\left(0, {\alpha_i}^{-1}\right). \tag{2.7}$$

The key to obtain sparsity is the use of independent hyperparameters $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_N)^T$,

one per weight (or basis function), which moderate the strength of the prior information.

Using Bayes' rule and having the prior distribution and likelihood function (2.7 and 2.6), the posterior distribution over the weights would be a multivariate Gaussian distribution:

$$p(\mathbf{w} \mid \mathbf{t}, \boldsymbol{\alpha}, \sigma^2) = \frac{p(\mathbf{t} \mid \mathbf{w}, \sigma^2)p(\mathbf{w} \mid \boldsymbol{\alpha})}{p(\mathbf{t} \mid \boldsymbol{\alpha}, \sigma^2)} = N(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{2.8}$$

where the covariance and the mean are:

$$\boldsymbol{\Sigma} = (\sigma^{-2}\boldsymbol{\Phi}^T\boldsymbol{\Phi} + \mathbf{A})^{-1}, \tag{2.9}$$

$$\boldsymbol{\mu} = \sigma^{-2}\boldsymbol{\Sigma}\boldsymbol{\Phi}^T\mathbf{t}, \tag{2.10}$$

and $\mathbf{A} = diag(\alpha_1, ..., \alpha_N)$.

The likelihood distribution over the training target $\mathbf{t}$, given by (2.6), is marginalized with respect to the weights to obtain the marginal likelihood for the hyperparameters:

$$p(\mathbf{t} \mid \boldsymbol{\alpha}, \sigma^2) = \int p(\mathbf{t} \mid \mathbf{w}, \sigma^2)p(\mathbf{w} \mid \boldsymbol{\alpha})d\mathbf{w} = N(0, \mathbf{C}), \tag{2.11}$$

where the covariance is given by $C = \sigma^2\mathbf{I} + \boldsymbol{\Phi}\mathbf{A}^{-1}\boldsymbol{\Phi}^T$. Values of $\boldsymbol{\alpha}$ and σ^2, which maximize (2.11), cannot be obtained in closed form, thus the solution is derived via

an iterative maximization of the marginal likelihood $p(\mathbf{t} \mid \boldsymbol{\alpha}, \sigma^2)$ with respect to $\boldsymbol{\alpha}$ and σ^2:

$$\alpha_i{}^{new} = \frac{1 - \alpha_i \Sigma_{ii}}{\mu_i{}^2}, \tag{2.12}$$

$$(\sigma^2)^{new} = \frac{\|\mathbf{t} - \boldsymbol{\Phi}\boldsymbol{\mu}\|}{N - \sum_{i=1}^{N}(1 - \alpha_i \Sigma_{ii})}. \tag{2.13}$$

The basic RVM algorithm iterates over (2.9), (2.10), (2.12), and (2.13), reducing the dimensionality of the problem when α_i is larger than a threshold (note that α_i has a negative power in (2.7)). The algorithm stops when the likelihood $p(\mathbf{t} \mid \boldsymbol{\alpha}, \sigma^2)$ stops increasing. The non-zero elements of $\mathbf{w}$ are called Relevance Values, and their corresponding data points are called Relevance Vectors (RVs) as an analogy to the Support Vector Machine. Having the relevance vectors, $\{\mathbf{x}_r\}_{r=1}^{|RVs|}$, and the relevance values, $\{w_r\}_{r=1}^{|RVs|}$, the RVM makes prediction on a new data instance $\mathbf{x}_*$:

$$y_* = \sum_{r=1}^{|RVs|} w_r K(\mathbf{x}_*, \mathbf{x}_r), \tag{2.14}$$

where $|RVs|$ denotes the cardinality of the set of relevance vectors.

2.3.2 Relevance Vector Classification

In the binary classification framework, each target t_i is binary (either 0 or 1). In this case, the model is assumed noise-free, i.e., $\sigma^2 \equiv 0$. Applying the sigmoid function $\rho(y) = 1/(1 - e^{-y})$ to $y(\mathbf{x}; \mathbf{w})$, and adopting the Bernoulli distribution for

$p(t_i \mid \mathbf{x}_i, \mathbf{w})$, we can rewrite the likelihood as:

$$p(\mathbf{t} \mid \mathbf{w}) = \prod_{i=1}^{N} \rho(y(\mathbf{x}; \mathbf{w}))^{t_i}(1 - \rho(y(\mathbf{x}; \mathbf{w}))^{1-t_i}. \tag{2.15}$$

However, here we cannot obtain the marginal likelihood analytically similar to the regression case. Therefore, we can use an iterative procedure [91] which involves repeatedly solving these two coupled problems: (i) An optimization of a regularized logistic model: From $\log\{p(\mathbf{w} \mid \mathbf{t}, \boldsymbol{\alpha})\}$, we find the most probable weights $\mathbf{w}_{MP}$ for the current fixed values of $\boldsymbol{\alpha}$; (ii) A regression RVM: Computing the Hessian of $\log\{p(\mathbf{w} \mid \mathbf{t}, \boldsymbol{\alpha})\}$ at $\mathbf{w}_{MP}$, then the covariance $\boldsymbol{\Sigma}$, and correspondingly, updating $\boldsymbol{\alpha}$.

2.3.3 RVM versus SVM

The RVM is a probabilistic model whose functional form is equivalent to the SVM [79] and achieves comparable recognition accuracy to it [12]. The SVM is another sparse method for training a model such as (2.5), where the basis functions have the form of kernel functions. The SVM expresses predictions in terms of a linear combination of kernel functions centred on a subset of the training data called Support Vectors (SVs). Given labelled training data, the SVM algorithm outputs an optimal hyperplane which categorizes new examples, that is, it makes predictions on examples that are not part of the training set. The optimal hyperplane is a hyperplane that maximizes the margin of the training data (see Figure 2.2). To consider training errors, the SVM also uses a soft margin assumption which allows a

few training data to fall on the wrong side of the hyperplane. The optimal hyperplane is specified by the support vectors. Support vectors are samples that are closest to the hyperplane. However, the relevance vectors in the RVM, unlike the SVM, tend to represent more prototypical examples rather than data points close to the decision boundary [92].

Some advantages of the RVM over the SVM are discussed in Tipping [92]: (i) In the SVM, it is necessary that we tune the parameters related to the cost of the soft margin (i.e., misclassification penalty). However, analogous parameters in the RVM are automatically estimated by the algorithm. (ii) In the SVM, the kernel function must satisfy Mercer's condition (which says the square kernel matrix should be Positive Definite Symmetric), while the RVM framework allows the use of an arbitrary kernel function. (iii) The RVM method, unlike the SVM, is based on a Bayesian inference, and provides probabilistic outputs. (iv) The RVM provides much sparser models in terms of the number of examples (i.e., relevance vectors) than the SVM (i.e., support vectors).

A disadvantage of RVM over the SVM is that the RVM can get stuck in a local optimum. However, the convex optimization algorithm employed by the SVM guarantees to find a global optimum. Another disadvantage of RVM is hidden in calculating the covariance (2.9) which requires a matrix inversion. The issue is that if this matrix is ill-conditioned, then its inverse will be prone to large numerical errors (or not available if it is singular). A matrix is ill-conditioned if the condition number

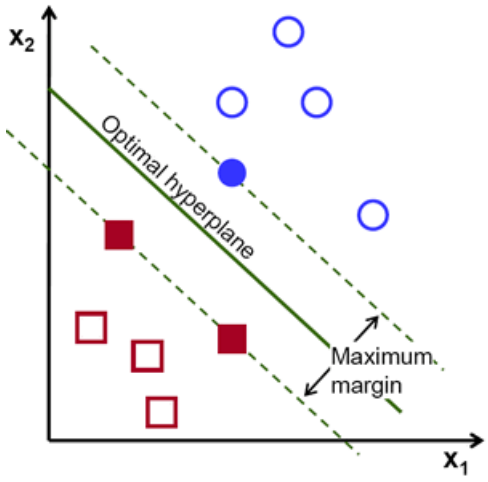


Figure 2.2: An example of a linear SVM. Support vectors are shown with the solid shapes.

is too large (and singular if it is infinite) [97]. The condition number of a matrix is defined as $cond(\mathbf{A}) = \|\mathbf{A}\| \cdot \|\mathbf{A}\|^{-1}$.

2.4 Evaluation Measures and Techniques

In the process of constructing a classification or regression model, we first split the full labelled data into training and test samples. The training sample is used for model selection and training, while the test sample is used to show the overall performance of the trained model on unseen instances. In this section, we describe some measures and techniques which are used or mentioned in this research for evaluating the performance of a regression or binary classification model.

Sensitivity, Specificity, and Accuracy

There are three main statistical measures for expressing the estimated performance in a binary classification when evaluating the classifier on the testing data:

- Sensitivity (SN), measures the accuracy of positive classifications, and is defined as $SN = TP/(TP + FN)$, where TP and FN refer to the number of true positives and the number of false negatives, respectively.

- Specificity (SP), measures the accuracy of negative classifications, and is defined as $SP = TN/(TN + FP)$, where TN and FP refer to the number of true negatives and the number of false positives, respectively.

- Accuracy (ACC), represents the proportion of true results, both positive and negative, in the selected population, and is defined as $ACC = (TP+TN)/(TP+FP+TN+FN)$.

A good test is a one who has both high SN and SP, and consequently, high ACC. However, sometimes because of specific conditions in the population of interest (e.g., imbalanced datasets), the estimated performance of a classifier may result in high accuracy, but large difference in absolute value between SN and SP. Therefore, ACC needs to be interpreted cautiously.

Correlation Coefficient

In this thesis, we will compute the correlation coefficient between the observed and predicted values primarily in regression. Pearson's Correlation Coefficient (PCC, or R) is a measure of linear dependence between two variables X and Y:

$$R_{xy} = \frac{\sum_{i=1}^{n} X_i Y_i - \frac{\left(\sum_{i=1}^{n} X_i\right)\left(\sum_{i=1}^{n} Y_i\right)}{n}}{\sqrt{\left(\sum_{i=1}^{n} {X_i}^2 - \frac{\left(\sum_{i=1}^{n} {X_i}^2\right)^2}{n}\right)\left(\sum_{i=1}^{n} {Y_i}^2 - \frac{\left(\sum_{i=1}^{n} {Y_i}^2\right)^2}{n}\right)}}$$

where n refers to the number of pairs of data. The correlation coefficient shows that how closely one variable is related to another variable. The value of R falls in the range [-1,1]. A correlation coefficient of 0 means that there is no linear relationship between the two variables. The closer a correlation coefficient to +1 or -1, the stronger the relationship is between variables. In this case of regression, if an X_i is the observed output value for an input, its corresponding Y_i represents the predicted output value for that input by a predictor. In this way, a PCC close to 1 represents that a predictor is making accurate predictions for a regression problem.

Coefficient of Determination

Coefficient of determination (R^2) is the square of correlation coefficient. It is a statistical measure that indicates the proportion of the variance in the dependent

variable y that is predictable from the independent x-variables. The range is 0 to 1. The coefficient of determination gives us an idea of how many data points fall within the results of the fitted regression line. In general, the higher the (R^2), the better the model fits our data.

Depending on the situation during the thesis, we may use either of correlation coefficient or coefficient of determination measures in evaluating a regression model. More precisely, to compare our results with a related work which uses a specific measure, we will also evaluate our method with the same measure. The goal is not to compare results between datasets (e.g., yeast versus flax) but to compare between previous results on the same dataset.

Cross-validation

In learning a prediction function, partitioning the available data into three sets - training, validation and test sets - is a reasonable solution to avoid overfitting (i.e., overfitting as the result of model selection based on training and testing on the same data). However, by partitioning the data into three sets, we drastically reduce the number of samples which can be used for learning the model. Also, the results depend on a particular random selection of the pair (train, validation) sets. Cross-validation is a solution to this problem. Cross-validation is a model evaluation method for estimating the performance of a predictive model. It is also used for model selection (tuning the free parameters of the algorithm).

In a k-fold cross-validation, we partition the original dataset into k equal-sized subsets, randomly. Then, we train our model k times, each time leaving one of the k subsets for testing, and using all the remaining $k-1$ subsets for training. In this way, each of the k subsets is used exactly once as the test set:

$$dataset = \bigcup_{j=1}^{k} d_j,$$

$$Fold_i : \begin{cases} testset = d_i \\ \\ trainset = dataset - d_i \end{cases} \quad , \quad i = 1, ..., k.$$

The k results (i.e., computed prediction measures) from the folds can be averaged to produce the final evaluation result.

As a cross-validation result depends on how a given dataset is divided into folds, sometimes a repeated cross-validation is applied at an added cost. In a repeated k-fold cross-validation, k-fold cross-validation runs for multiple times (e.g., N) using different split into folds, and the result will be the average of all N cross-validation results. Repeated k-fold cross-validation allows to get a more precise estimate of the expected predictive performance than non-repeated k-fold cross-validation.

ROC Curve Analysis

A Receiver Operating Characteristic (ROC) curve is a commonly used way to visualize the performance of a binary classifier. In a ROC curve, the true positive

rate (or, SN) is plotted against the false positive rate (or, $1-SP$) for different cut-off points of a parameter. A typical ROC curve is shown in Figure 2.3. Generally, ROC curve analysis is undertaken to:

1. Assess the overall discriminatory ability of a binary classifier, i.e., the ability of a classifier to correctly classify the samples into two groups, positive and negative.

2. Recognize the best SN and SP. As a result, a ROC curve helps in determining the best cut-offs that maximize SN and minimize $1 - SP$.

The Area Under the Curve (AUC) is a common metric to compute the strength of a classification: An AUC of the ROC curve close to 1 indicates a strong test, and an AUC close to 0.5 (i.e., random classifier performance) represents a weak test.

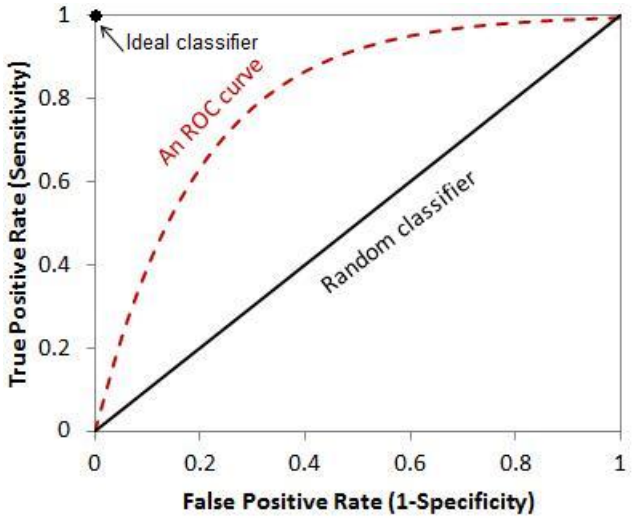


Figure 2.3: A typical ROC curve.

Chapter 3

Methodology

In this chapter, we introduce separate definitions for two different forms of the sparse Bayesian learning, i.e., Kernel RVM and Basis RVM, and explain our proposed ensemble method, Ensemble RVM, for ranking purposes.

3.1 Kernel RVM versus Basis RVM

In this research, we define sparse Bayesian learning in such a way that we can discriminate between kernel and basis functions, i.e., "kernel" RVM versus "basis" RVM. For example, we define two types of linear RVMs, which we call linear kernel RVM and linear basis RVM. In a linear kernel RVM, the basis functions in (2.5) are

linear kernel functions, i.e.,

$$\phi_m(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}_m) = \langle \mathbf{x}, \mathbf{x}_m \rangle.$$

When we use linear kernels, in fact we have no mapping. In other words, there is no feature space (as we use input vectors directly), so our estimator tries to pass a hyperplane through input vectors in the input space (e.g., in the regression case).

In our linear basis RVM, the basis functions are linear and equal to the features of the input vectors, i.e.,

$$\phi_m(\mathbf{x}) = \mathbf{x}^{[m]},$$

where $\mathbf{x}^{[m]}$ refers to the m-th feature in an input vector $\mathbf{x}$ with M dimensions. We can view it as if we have no basis function in a linear basis RVM, as we use input vectors directly in (2.5) instead:

$$y(\mathbf{x}; \mathbf{w}) = \mathbf{w}^T \mathbf{x}.$$

Therefore in (2.6), $\mathbf{w} = (w_0, w_1, ..., w_M)^T$ are weights, where M is the number of

features, and the design matrix

$$\mathbf{\Phi}_{N\times(M+1)} = \begin{bmatrix} 1 & \mathbf{x}_1{}^{[1]} & & \mathbf{x}_1{}^{[M]} \\ 1 & \mathbf{x}_2{}^{[1]} & & \mathbf{x}_2{}^{[M]} \\ & & \cdots & \\ 1 & \mathbf{x}_N{}^{[1]} & & \mathbf{x}_N{}^{[M]} \end{bmatrix}, \tag{3.1}$$

where the first column handles the intercept w_0, and N is the number of training individuals.

Thus, this linear-basis RVM will find the RVs which correspond to the features; i.e., the obtained sparsity will be in the feature set rather than the training individuals. This is exactly what we expect from a feature selection method. Therefore, this RVM can perform target prediction as well as feature selection. For example, in a genomic selection/GWAS in crop breeding, the individuals are breeds of a crop, the features are the markers, and a phenotype is a target. Then, a linear basis RVM would identify a subset of relevant markers to that phenotype, while it is trained for phenotype prediction.

We should note that there is not an SVM counterpart for a basis RVM, as the design matrix (3.1) resembles a non-PDS function which specifically cannot be used in an SVM. In Chapters 5 through 7, we use linear basis RVMs to rank which markers contribute most to a prediction.

3.2 Ensemble RVM

An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way to classify examples [25]. Ensembles often produce better predictive performance than a single model by decreasing variance (bagging), bias (boosting), or improving predictions (stacking) [106]. Moreover, ensemble techniques have the advantage of handling large data sets and high dimensionality because of their divide-and-conquer strategy. Random Forests [16] and Gradient Boosting Machines [33] are examples of ensemble methods.

In this research, we employ ensemble RVM with the bagging approach. Bagging stands for bootstrap aggregating [15]: base models are trained on bootstrap subsamples of the training set and their predictions are aggregated through majority voting or averaging. Bootstrapping [27] is a type of resampling where large numbers of smaller samples of the same size are repeatedly drawn, with replacement, from an original sample. Bagging is commonly used as a resolution for the instability problem in estimators. Bagging has been used in the random forest algorithm.

We use ensemble RVM for target prediction (in regression or classification case) and feature selection. This is the contribution of the thesis. Specifically, for feature selection purposes, we construct ensembles composed of linear basis RVMs, while for target prediction we mainly use kernel RVMs. Each RVM in an ensemble finds a set of representatives or RVs which represent important training individuals or features, depending on the type of RVMs in the ensemble. Then, aggregating RVs of the

ensemble lets us rank the individuals or features. Ranking mechanisms allow us to reduce dimensionality and enhance generalization [75]. Furthermore, they enable us to recognize interpretable or insightful individuals/features in the model. This is the attribute that we can benefit from in some applications such as genomic selection and GWAS, as we will see in the following chapters.

As a sample, a pseudo-code for training an ensemble of basis RVMs in regression is shown in Algorithm 1. This ensemble is used to rank the features. Herein we omitted any processes related to the model selection phase. In other words, we assumed that the number of RVMs in the ensemble, *numRVMs*, and the bag size, *bagSize*, are known in advance. Therefore, we showed both the training and the test phases for the ensemble in the same loop. The inputs of the algorithm are the training and test datasets, that each of which include a design matrix and a vector of real-valued labels. The algorithm outputs the ranks of the features in the training dataset and the performance (PCC) of the ensemble over the test dataset. As the algorithm shows, each RVM in the ensemble is trained with a *bag* of the training dataset. As we are in the regression case, we assume the labels are noisy, so we assign a Gaussian distribution to the likelihood and consider a standard deviation for the noise (e.g., 0.1) in the main RVM algorithm, which we have explained in Section 2.3.1. Also, we should set the number of iterations in the RVM algorithm. After training an RVM with *SparseBayes*, the trained RVM gives us its inferred weights and RVs. Then, we update the rank vector according to the identified RVs. The rank vector is a vector

that counts the number of times each vector is an RV in any RVM in the ensemble. Having the trained RVM, we can also obtain the labels of the test individuals. After training all the RVMs in the ensemble, we will have the feature ranks, and calculate the ensemble vote on the test individuals by averaging the result of the RVMs.

Algorithm 1: EnsembleBasisRVR

Input: *trainData, testData*

Output: *rankVect, testPCC*

Initialization:

$rankVect \leftarrow zeroVect$

$testPCC \leftarrow zero$

$ySumVect \leftarrow zeroVect$

for $i = 1$ **to** $numRVMs$ **do**

$bag \leftarrow Bootstrap(trainData, bagSize)$

$weightVect, RVs \leftarrow SparseBayes(likelihood = \text{“}Gaussian\text{”}, noisSTD = 0.1,$
$iterations = 2000,$
$designMat = bag.designMat,$
$labelVect = bag.labelVect)$

$Update(rankVect, RVs)$

$yVect \leftarrow testData.designMat * weightVect$

$ySumVect \leftarrow ySumVect + yVect$

$ensembleVoteVect \leftarrow Average(ySumVect, numRVMs)$

$testPCC \leftarrow Correlation(testData.labelVect, ensembleVoteVect)$

3.3 Algorithmic Complexity

Herein we discuss the computational complexity of a single RVM or an ensemble RVM in details. We should note that the complexity of a predictor differs in the training phase versus the prediction phase.

We exclude the model selection phase, and we suppose that we have a high dimensional dataset with N training individuals, each with p features ($p \gg N$). During the training phase, first we should construct the kernel matrix in a kernel RVM, or the design matrix in a basis RVM. Constructing a design matrix for a basis RVM with linear basis functions (our case in this research) can be considered to take constant time. However, computing a numerical kernel matrix is quadratic in the number of training individuals, $O(N^2)$, as we should calculate a dot product for every two pairs of individuals. Also, the complexity of computing an n-gram rational kernel over a pair of strings is linear in the sum of the length of the strings [3]. In other words, it is is equal to the length of the longest sequence, or p in our case. Therefore, the complexity of constructing an n-gram kernel matrix would be $O(N \times p)$.

Given the kernel or design matrix, we enter the main part of the training an RVM. Based on the sparse Bayesian learning algorithm 2.3, the computationally intensive part of the RVM algorithm is the matrix inversion in (2.9) which requires $O(N^3)$ operations for a dataset of size N in a kernel RVM [81]. Also, the space complexity is $O(N^2)$ for the memory requirements of $\mathbf{\Phi}$ and $\mathbf{\Sigma}$ matrices [81]. Similarly, if we have a basis RVM, the algorithm requires $O(N \times p)$ storage and $O(p \times N^2)$ computation.

Therefore, repeatedly computing and inverting the Hessian matrix is the most costly part in the training phase of an RVM.

The complexity of training an ensemble can be viewed identical to a single RVM, as the number of RVMs in an ensemble is often much less than the number of training individuals (or features). Also, taking a bag of the training data in each RVM member of an ensemble is linear in the bag size, which is at most equal to N in a kernel RVM (or p in a basis RVM); thus, it does not increase the overall computational complexity.

In the prediction phase, the computational complexity of an RVM is linear in the number of RVs in the RVM, $O(|RVs|)$, based on (2.14). As the number of RVs is sparse, the trained RVM is fast in prediction. Consequently, the computational complexity of prediction in an ensemble RVM is $O(|RVs|_1 + ... + |RVs|_{numRVMs})$, which will be linear in N in an ensemble of kernel RVMs (or p in an ensemble of basis RVMs) in the worst case.

3.4 Implementation Tools

To do our experiments on the datasets in Chapters 5,6 and 7, we use the following machine learning tools:

- To implement the RVMs, we use SpareBayes software package for Matlab [90].
- For computing rational kernels (n-gram and composition kernels), we use Open-

FST (Open Finite-State Transducer) library [2, 5] and OpenKernel library [4].

- For computing numerical kernels and other machine learning tasks such as implementing an SVM, we use Scikit-learn package in Python [71].

Chapter 4

Related Work

We considered the genomic selection problem as a regression (or as a classification) problem, and genetic marker association of a complex trait as a feature selection problem. To date, there has not been any research on the application of RVM in genomic selection or genome-wide association study, either in plants or animals. However, there is research on RVM in some other bioinformatics applications such as identifying particular motifs in a sequence [26, 58] and predicting biological networks [6, 99]. Also, there is research on feature selection techniques based on machine learning methods in a few applications in bioinformatics such as microarray analysis [44]. In this chapter, first we present an overview of the genomic selection problem. Then, we review the existing research works on RVM-based applications in bioinformatics. Next, we explain genome-wide association study and some related classical methods for analysing complex traits. Last, we review some latest research

works on feature selection techniques in bioinformatics applications.

4.1 Genomic Selection

Genomic selection involves predicting a phenotype (e.g., traits, disease risk) based on all available markers across the entire genome, whereas traditional marker-assisted selection attempts to identify individual loci in a genome significantly associated with a trait. Genomic selection has been applied to animal, plant, and human species. Meuwissen et al. [61] introduced the concept of genomic selection, versus the marker-assisted selection process. As genomic selection uses all marker data as predictors of performance, it will consequently deliver more accurate predictions [48]. A genetic marker is a variation in DNA that can be used to differentiate between individuals or species. It can be a short DNA sequence, such as Single Nucleotide Polymorphism (SNP), which is a single base-pair change, or a long one, such as Single Sequence Repeat (SSR).

Machine learning methods have had many contributions in genomic selection research. To name a few applications, see [41, 54] in human, [13, 39, 40, 48, 56, 70] in plant, and [37, 65, 103] in animal research. For example, Guo et al. [41] applied logistic regression model, support vector machines, and gradient boosted trees for predicting *Anorexia nervosa*[1] disease risk, having genotype data of 3940 Anorexia cases and 9266 controls. Li et al. [56] assessed several models such as Bayesian meth-

[1] *Anorexia nervosa* is a complex psychiatric eating disorder with a genetic contribution.

ods, support vector regression, and random forests, in predicting the flowering time traits of *Brassica napus*[2] across 1674 genotyped SNPs under ten different natural environments and three geographical regions. Yao et al. [103] applied a semi-supervised support vector regression algorithm to predict residual feed intake in dairy cattle with a small reference population, having a dataset of 3792 cows with or without measured phenotypes. In all of these applications, the central problem is to predict an organism's phenotype from knowledge of its genotype and environment.

Genomic selection in plants and animals have been mainly used in breeding. In this case, genomic selection is based on two distinct and related groups: training and breeding populations [24]. The training population, which is both genotyped and phenotyped, is used to train a learning model; and the breeding population is a set of individuals that are genotyped but not phenotyped. The trained model is then used to predict breeding or genotypic values of non-phenotyped selection candidates.

Heritability

Heritability is a concept used in the fields of breeding and genetics that describes the degree of variation in a phenotypic trait in a population that is due to genetic variation in that population [98]. Heritability is relative to specific population in a particular environment since contribution of genetic factors is relative to contribution of other factors such as environment. Its number can range from 0 (no genetic

[2] *Brassica napus* or Rapeseed is a source of vegetable oil.

contribution) to 1 (all differences on a trait reflect genetic variation). Two specific types of heritability are broad-sense heritability and narrow-sense heritability. The broad-sense heritability (H^2) is the variance in the phenotype measurements (V_P) due to genetic factors (V_G):

$$H^2 = \frac{V_G}{V_P}.$$

Phenotypic variance combines the genetic variance with the environmental variance. Genetic variance usually has three major components: the additive variance (V_A), dominance variance (V_D), and epistatic variance (V_J):

$$V_G = V_A + V_D + V_J.$$

Additive genetic effects are the contributions to the final phenotype from more than one gene, or from alleles [3] of a single gene, that combine in such a way that the sum of their effects in unison is equal to the sum of their effects individually. Non-additive genetic variation [86] results from interactions between genes. Interactions between genes at the same locus [4] are called dominance, and interactions between genes at different loci are called epistasis. In addition to the three major genetic effects, there may be parental imprinting effects [77]. Imprinted genes are genes whose expression is determined by the parent that contributed them. Imprinted genes do not follow the

[3] An allele is one of the possible forms of a gene.

[4] A locus in genetics is a fixed position on a chromosome, like the position of a gene or a marker.

usual rule of inheritance that both alleles in a heterozygote are equally expressed [51]. The narrow-sense heritability (h^2) is the variance in the phenotype measurements due to additive genetic factors:

$$h^2 = \frac{V_A}{V_P}.$$

Data Representation

Input: There is not a unique way to represent a sequence of SNPs in a vector space for applying in a machine learning method. Typically, an individual in a genomic selection problem is a fixed-length sequence of biallelic SNPs wherein an allele is coded either as 1 or 2. For example, consider this sequence with five biallelic SNPs:

$$1\ 2 \quad 2\ 2 \quad 1\ 2 \quad 2\ 1 \quad 1\ 1,$$

where each of the pairs represents the value in both the mother's and father's chromosomes in an individual, e.g., the first pair 1 2 indicates that in the first locus in this offspring genome, the inherited allele from the father and the mother is 1 and 2, respectively. One way of representing the above example as a numerical vector, applicable in a numerical kernel function, can be:

$$[0 \quad -1 \quad 0 \quad 0 \quad +1],$$

where 1 1 coded as $+1$, 2 2 as -1, and 1 2 or 2 1 as 0. To be applicable in a sequence kernel function, we can represent it as a string:

$$CBCCA,$$

where 1 1 is coded as A, 2 2 as B, and 1 2 or 2 1 as C. Obviously, the representation method will affect on the prediction result and performance, apart from what classifying method we choose.

Output: A typical genomic selection classifier predicts either a binary trait or a quantitative trait. Quantitative traits are real-valued numbers (e.g., grain yield in kg/ha). However, binary traits only have two distinct phenotypic values (e.g., affected/unaffected), so we can represent them as $-1/+1$ or $0/1$ output values.

4.2 RVM Applications in Bioinformatics

In the following, we discuss existing research that exploits RVMs for bioinformatics applications for prediction purposes, with an emphasis on the kernel type that each RVM-based solution used.

Non-sequence kernel RVM: Down and Hubbard [26] introduced an RVM model with an incremental training procedure for classifying and detecting interesting individual points and regions in sequences. They defined several basis functions based on the occurrence of particular motifs within the sequence, or more precisely, position-weight matrices[5]. These basis functions were designed to be used in the RVM implementations, for purposes such as detection of transcription start sites, splice sites, or small motifs. However, the authors did not report any results of applying their

[5]This is a matrix where each element represents the probability of a given nucleotide (or amino acid) occurring at a particular position in the sequence.

RVM model in a problem despite stating that their model was successful.

Sequence kernel RVM: Li et al. [58] used an RVM classifier to identify a small subset of promoter motifs that have regulatory functions in glucose-regulated and ABA-regulated genes in *Arabidopsis*. Having the promoter sequences of variable lengths as input, the RVM classifier gives an estimate of the probability that the associated gene is up-regulated or down-regulated as output. The authors constructed RVMs for classifying glucose-regulated and ABA-regulated genes with these datasets as inputs: a collection of ∼1700 glucose and ∼1300 ABA up- and non/down-regulated promoters. In this RVM model, a basis function $\phi_i(\mathbf{x})$ represents the number of times an arbitrary substring s_i, such as a 5-mer [6], occurs in a promoter sequence $\mathbf{x}$. Li et al. used 10-fold cross-validation to estimate the performance of the model. Using the AUC as measure, they demonstrated that the RVMs have a classification rate of ∼70%. The authors could also validate the top-weighted promoter sequences selected by the RVM strategy, based on the functional knowledge of known promoter motifs. Li et al. established a detailed model of glucose and ABA transcriptional regulatory networks and their interactions, having the predicted promoter motifs. Their research shows that although the RVM did not show improvement in classification accuracy (compared with some related work which used probabilistic methods), the sparse feature selection attribute of the RVM still could help in identifying significant motifs, given a sufficiently rich set of input sequence features.

[6] The term k-mer refers to all the possible substrings of length k that are contained in a string.

Ensemble of non-sequence kernel RVMs: Wu et al. [99] used an ensemble method for prediction of human functional genetic networks from heterogeneous data. Large-scale datasets and massive missing data values are sources of major problems that we confront in constructing genetic networks from heterogeneous data sources. To tackle these problems, Wu et al. proposed a combination of AdaBoost, RVM, and reduced-feature model. AdaBoost [78], or Adaptive Boosting, is an ensemble learning algorithm which constructs a strong classifier from a combination of simple weak classifiers, which were RVMs in this research. Using the reduced-feature modelling, Wu et al. considered a set of base models which used subsets of the complete feature sets (all of the different sources of the data). Then, they trained each base model using RVM-AdaBoost (i.e., first level of ensemble); and at the end, they generated an ensemble of all base models through averaging of the outputs from all base models as the final output (i.e., second level of ensemble). Wu et al. defined four kernels (called KC1-4) using different combination of radial basis kernel (see Section 2.2) and diffusion kernel [52]. A 10-fold cross validation on training set showed $\sim$85% performance (using AUC) for the kernel KC1. The authors also demonstrated that the sparseness of RVM-based ensemble model is able to significantly reduce the prediction time, which is crucial in such a large-scale problem. Their research is the only one which adopted an RVM-based ensemble for an application in bioinformatics.

Non-sequence kernel RVM: An et al. [6] proposed a method based on the RVM model combined with Local Phase Quantization (LPQ), described below, to predict

protein-protein interactions from protein sequences. They used their method to predict interactions in human and yeast datasets with ∼11000 and ∼8000 protein pairs, respectively. In their proposed method, first each protein sequence is represented as an $M \times 20$ position-weight matrix, wherein M is the length of the protein sequence and 20 corresponds to twenty amino acids. As the obtained matrices have a variable number of rows, An et al. used the LPQ method to convert each matrix to a 256-dimensional feature vector. Originally, LPQ is an image processing operator that is used to process spatial blur in textural features of images. After this stage, the authors used Principal Component Analysis (PCA), which is a procedure for identifying a smaller number of linearly uncorrelated variables, to convert 256 features into 180 features in human and 172 in yeast. Then, they used a Gaussian kernel RVM for classification of protein pairs to interacting/non-interacting. Using a 5-fold cross-validation, An et al. [6] achieved a high accuracy of 97.62% in human and 92.65% in yeast compared to the other methods including SVM (combined with LPQ, similar to their RVM model) which showed at least 10% difference in prediction accuracies. In this research, the sparseness of the model was not important and only classification strength of the RVM was presented. The authors concluded the reasons for the better performance of the RVM model, particularly compared to SVM, are less computational work of the kernel function, and also no obligation to satisfy Mercer's condition. However, they did not clarify how these attributes may result in a higher accuracy.

4.3 Genome-Wide Association Study

A Genome-Wide Association Study (GWAS), is an observational study of a genome-wide set of genetic variants in different individuals to see if any variant is associated with a trait. GWAS is a relatively new way [7] for scientists to discover genetic factors underlying phenotypic variation. The most common approach of GWAS is the case-control design which is appropriate for binary traits. For example, scientists can search the genome for SNPs which occur more frequently in people with a particular disease than in people without the disease (e.g., [68, 96]). For quantitative traits, such as height and yield in plants, there are several variations to the case-control approach (see [17, 53] for more information).

The traditional alternative approach to GWAS is Quantitative Trait Locus (QTL) mapping/analysis. A QTL is a chromosomal region or genetic locus which correlates with variation in a phenotype (the quantitative trait). QTL analysis is a statistical method that links two types of information, phenotypic data (trait measurements) and genotypic data (molecular markers), in an attempt to explain the genetic basis of variation in complex traits. To begin an analysis, scientists need to have parental strains (two or more) that differ genetically for the trait of interest. Then, the parental strains are crossed to create F1 individuals, which are then crossed using one of a number of different schemes (e.g., crossed among themselves) to create F2 individuals. Finally, the phenotypes and genotypes of the F2 population are

[7] The first successful GWAS was reported in 2005 [20].

scored. Markers that are genetically linked to a QTL influencing the trait of interest will segregate more frequently with trait values, whereas unlinked markers will not show significant association with phenotype (For more information on QTL analysis, see [63]).

To detect reliable phenotype–genotype associations, sometimes genetic diversity analysis is done along with an association study [35]. The genetic diversity observed within a population is a result of various evolutionary processes, such as mutation, recombination, and selection, that act on population. Genetic diversity serves as a way for populations to adapt to changing environments. When there is more variation, it is more likely that some individuals in a population possess variations of alleles that are suited for the environment. Genetic diversity of a population can be assessed by some simple measures such as the number of alleles per locus [8]. Also, some simulation software programs are commonly used to predict the future of a population given measures such as allele frequency and population size [45].

4.4 Feature Selection in Bioinformatics

We can view a GWAS as a feature selection problem wherein features are biological markers (e.g., SNPs). There are many studies on feature selection in bioinformatics, mostly applied on microarray gene expression data for classification [44]. Feature selection methods are generally used to reduce overfitting and to improve the accuracy and efficiency of learning algorithms, especially when there exist irrelevant

or redundant features. As the common attribute of many bioinformatics applications is high-dimensionality, such as microarray data analysis, feature selection methods are extensively used (For reviews on feature selection methods in bioinformatics see [44, 75, 95, 101]).

Among feature selection methods, ensemble learning, such as random forests and ensemble SVM, has had an increase in use due to their unique advantages in dealing with high-dimensionality. For instance, a recent work that uses ensemble SVM for gene selection and classification of microarray gene expression data, is the research done by Anaissi et al. [7]. Anaissi et al. used SVMs based on Recursive Feature Elimination (SVM-RFE) as the base classifiers of an ensemble. The SVM-RFE algorithm, proposed by Guyon et al. [42], trains a linear kernel SVM, removes the worst feature (the one with smallest weight value of the decision hyperplane given by the trained SVM), and then repeats the process with the rest of the features until all are exhausted. At the end, features are ranked according to when they were eliminated, with the most important eliminated last. Anaissi et al.'s approach employs ensemble and bagging methods, in a similar way as random forests, to improve overall feature selection and classification accuracy in SVM-RFE. In other words, Anaissi et al. constructs multiple SVM models at each iteration of SVM-RFE (the number of SVMs was set to 50 models based on the earlier work of Saeys et al. [76]). For evaluation, they applied both ensemble SVM-RFE and single SVM-RFE on a childhood leukaemia dataset with about 22,000 features and 60 samples (75%

of samples for training and 25% for testing). To select the most important features, they evaluated a range of different number of features (e.g., from 20 to 100) on the training set using cross-validation, and chose the one which gave the best AUC (i.e., equal to 36 features). Their experiments showed an average 9% better AUC accuracy in the ensemble architecture compared to single SVM-RFE.

Anaissi et al.'s approach and our proposed ensemble RVM method for feature ranking are similar in respect of adopting ensemble and bagging techniques. However, our method has advantages that the ensemble SVM-RFE lacks. A drawback of SVM-RFE is that it is computationally expensive since it goes through all features one by one and it does not take into account any correlation the features might have [44]. However in our approach, feature selection is embedded in RVM base learners and no iterative approach such as RFE is required, so there is no increase in computational complexity for this purpose. Also, we claim that the quality of feature subsets in our ensemble approach would be higher than its analogous ensemble SVM-RFE due to the different interpretations of support vectors and relevance vectors (see 2.3.3).

There are also limited studies that used different sparse Bayesian classification for finding disease-related genes in classifying gene expression data [18, 57, 102], though not mentioned in the above feature selection surveys. A most recent work done by Yang et al. [102] proposes a sparse Bayesian classification algorithm for selecting predictive features which are highly correlated, as in a biological process, multiple molecules are working together, resulting in correlated feature expression

levels. Yang et al. used an iterative convex optimization procedure for updating parameters and hyperparameters. Adopting this approach for updating allowed an efficient implementation of the algorithm via parallel computing. The authors showed the success of their method on simulated data with 500 samples and 50 features wherein there were 4 pairs of highly correlated features. Then, they applied their method on a public embryonal tumour gene expression dataset with 20 samples (balanced) and 5669 genes for classifying samples into two tumour types. Using a 10-fold cross-validation, they could find 98 features distinguishing tumours. Then, using the heatmap of correlation matrix, they demonstrated that the selected features are correlated. They also showed the detected gene ontology terms are consistent with the findings in a previous study.

Chapter 5

Experimental Results on Synthetic Data

5.1 Introduction

In this chapter, we investigate how RVMs perform on genomic selection and GWAS with a synthetic dataset. Before constructing models for real datasets, it is advantageous to experiment with the methods on synthetic data, as we can identify any possible obstacles or drawbacks in our methods. Also in this way, we can have an evaluation of our methods, as on synthetic data we know all true answers, i.e., phenotype values and most important markers, in advance. This opportunity will not happen when we work on real datasets in genomic selection or GWAS.

We show how ensembles of kernel RVMs perform in predicting quantitative and

binary traits (i.e., regression and classification) for genomic selection. Also, we will demonstrate how ensembles of basis RVMs identify important markers affecting the traits in a GWAS.

5.2 Dataset

We have done our experiments on a simulated dataset [88], produced for the 14th QTL-MAS workshop [87]. It consists of 3226 individuals in five generations, and each individual is genotyped for 10031 biallelic SNPs, arrayed on a genome encompassing five chromosomes. The first four generations contain 2326 individuals, phenotyped for two traits: a Quantitative trait (Qtrait) and a Binary trait (Btrait). The phenotypic values for the remaining 900 individuals, representing the fifth generation, are supposed to be predicted. Therefore, we considered the first four generations individuals (without pedigree information) as the training data, and the fifth generation individuals as the testing data. Other than the split into testing and training data, the pedigree information is not used.

In this simulated dataset, the Qtrait was determined by 37 QTL located on chromosomes 1-4: 30 additive (loci 1-30), 2 pairs of epistatic (loci 31-32, also loci 33-34), and 3 imprinting QTL (loci 35, 36 and 37). The Btrait was affected by a subset of 22 additive QTL determining the Qtrait. The QTL differed in the amount of their effect on the traits: there were two QTL with major effect on chromosome 3 (loci 14 and 17), a group of QTL with intermediate effect on chromosomes 1-3,

several QTL with small effect on chromosome 4, and no QTL on chromosome 5. Each simulated QTL was surrounded by 19-47 SNPs located within 1Mb distance from the locus.

In this dataset, an individual is a fixed-length sequence of biallelic SNP pairs wherein an allele is coded either as 1 or 2, and each pair represents the value in both the parents' chromosomes in an individual. For example, the pair 1 2 in the ith position of an offspring sequence indicates that in the ith locus in the offspring genome, the inherited allele from the mother and the father is 1 and 2, respectively. We represented an individual as a numerical vector such that 1 1 is coded as $+1$, 2 2 as -1, and 1 2 or 2 1 as 0.

5.3 Predicting Phenotypes

5.3.1 Predicting Quantitative Trait

Performance Measures

To assess predictive accuracy of the regression model, we used the Pearson Correlation Coefficient (PCC) and the mean squared error (MSE) of observed and predicted trait values.

Training size versus Performance

A training set is a fraction of all possible data that can approximately cover the whole data domain. In a problem domain such as ours, we might think that a learning method can learn better from a bigger set than a smaller one to discover useful relations and discard unnecessary ones. However, it is not favourable in RVM to have large sets, because of time and memory requirements. To investigate the stability of the RVM model and the influence of training size on the RVM performance, we ran some experiments. We randomly partitioned the original training data into a training set (85%) and validation set (15%) for 20 times. In each round, we iteratively (10 times) trained an RVM with bootstrap samples of training set with specific size, ranging from 100 to 2000 individuals, and calculated the PCC on the training part and the validation set. The results are shown in Figure 5.1. As the performance curve for the validation set shows, the RVM demonstrates instability when we increase the training size. In other words, a greater training size does not always imply a better performance (e.g. compare PCCs when $N = 1700$ and $N = 1900$).

As the dataset is a high-dimensional dataset (i.e., about 10000 features versus 2000 individuals), we considered that it is linearly separable, so we used a linear kernel in all of our predictors.

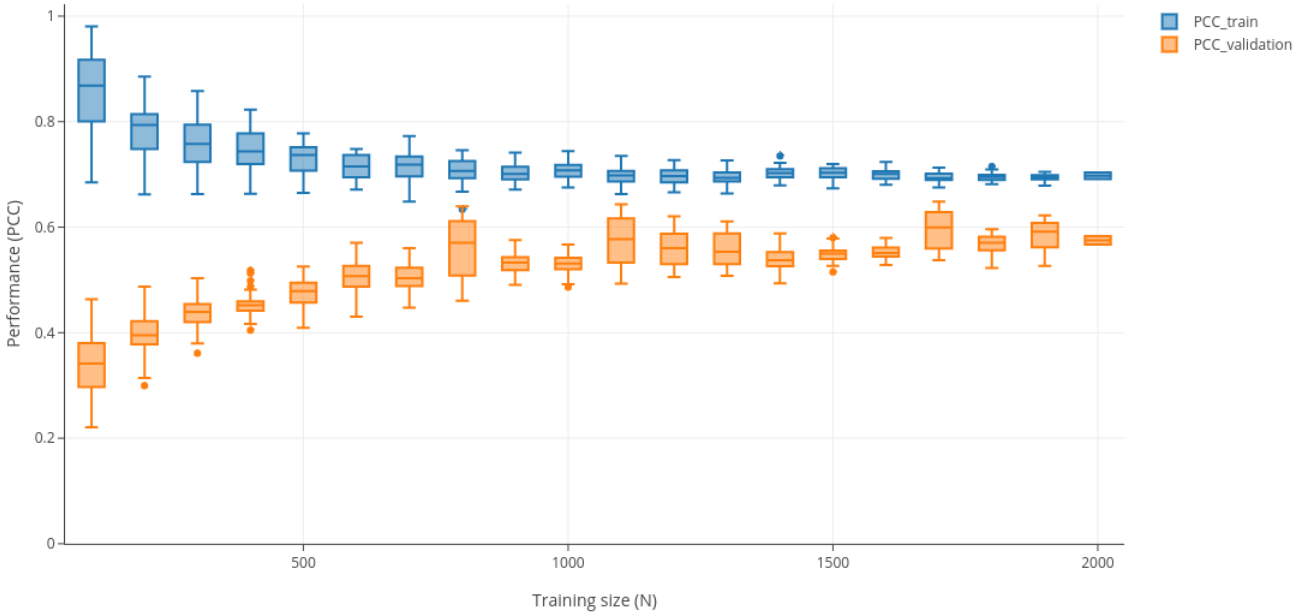


Figure 5.1: The impact of training size on the RVM performance (PCC) over the training and validation sets.

Ensemble Architecture

The RVM's instability issue, arisen from using different training sizes, led us to employ an ensemble RVM model to decrease the instability impacts. To find the best ensemble architecture, i.e., the number of RVMs in the ensemble and the size of training samples, we ran a set of experiments similar to the previous section: We repeatedly randomly partitioned the training data into training set and validation set. Then in each round, we iteratively trained a set of RVM ensembles, consisting of 5 to 100 RVMs, with bootstrap samples of training set in different sizes, and

calculated the PCC on the validation set. Figure 5.2 shows the curve of average PCC and MSE of an ensemble with 100 RVMs (other sizes not shown) over validation sets for different training sizes. As indicated in this figure, we will obtain the best PCC (i.e., 0.65), if the training set has 800 or 1100 samples. However, the absolute minimum MSE (i.e., 54.4) lead us to choose 1100 as the best training size for the ensemble. On the other hand, setting the number of RVMs in the ensemble with more than 40 did not show a noticeable improvement on the performance for this set of experiments. Nevertheless, we chose the largest number of RVMs for the ensemble (i.e., 100). In the situations in which we do not have any preferred option, using a larger ensemble might be more advantageous in the end, without adding up a considerable computational complexity to the training phase.

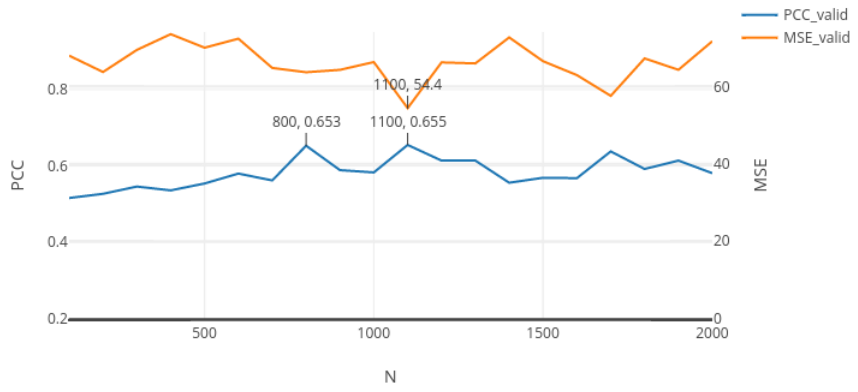


Figure 5.2: Average PCC and MSE over validation sets in different training size N in an ensemble with 100 RVMs.

Consequently, we constructed an ensemble with 100 RVMs and trained the en-

semble using bootstrap samples of 1100 items from the original training dataset. The PCC for the trained ensemble on the test data is 0.505 by averaging over RVMs.

About Relevance Vectors

We next consider the properties of the RVs in the ensemble. This will lead us to our new application of RVs for feature selection in Section 5.4. In one training of an ensemble of 100 RVMs using 1100 samples, there are approximately 1500 shared RVs. The RVs of an RVM is a sparse set of training individuals. In an ensemble, we can collect RVs of all RVMs and sort them based on the number of occurrences. Figure 5.3 demonstrates the Qtrait values for the most and least hit RVs (i.e., more than 29 and less than 2 occurrences, respectively). It is interesting that the top ranked RVs (blue marks) consist of the individuals with the highest and lowest trait values, while the lowest ranked RVs (orange marks) consist of the individuals in between. In fact, the top and bottom ranked RVs correspond to the maxima and minima of a function, respectively. We intuitively know that having extrema has greater importance than the other points in a regression problem. This confirms that RVs in the RVM represent more meaningful examples rather than data points close to the decision function as it is in the SVM.

Training Size versus Iterations

In the above experiments, we set the iteration parameter in the RVM algorithm to 5000. Almost 1% of the time, the RVM did not converge. Figure 5.4 shows RVM

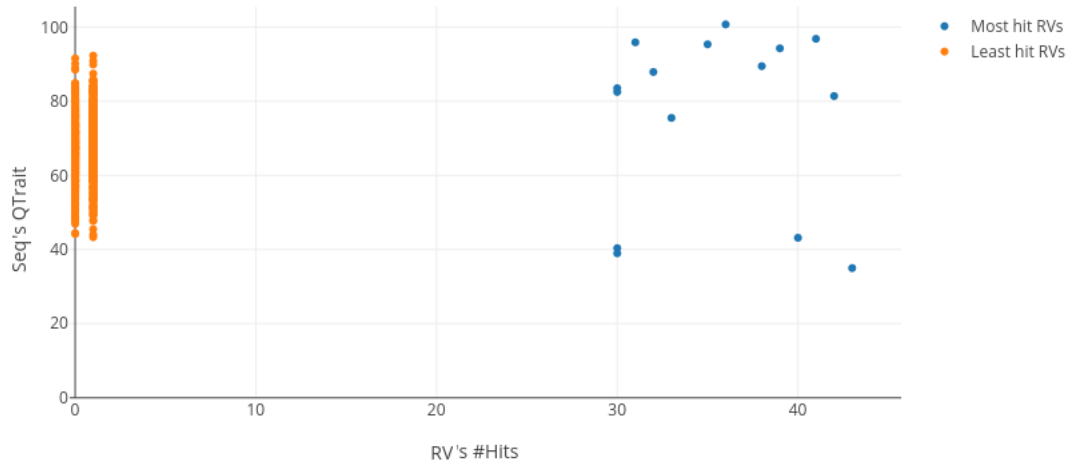


Figure 5.3: The range of Qtrait for the most and least hit RVs.

performance on training and validation sets versus different training sizes in both converged and non-converged RVMs. As the figure demonstrates, for some training sizes (e.g., less than 800), the algorithm always converges; and for others, the non-converged PCC is close to the converged PCC, if not better. It should be noted that the algorithm with less than 5000 iterations had more non-convergent cases (we examined down to 1500 iterations). Although non-convergent RVMs practically have similar performance in a set of experiments compared to their convergent counterparts, we raised the number of iterations to 5000 to have a smaller set of non-convergent cases.

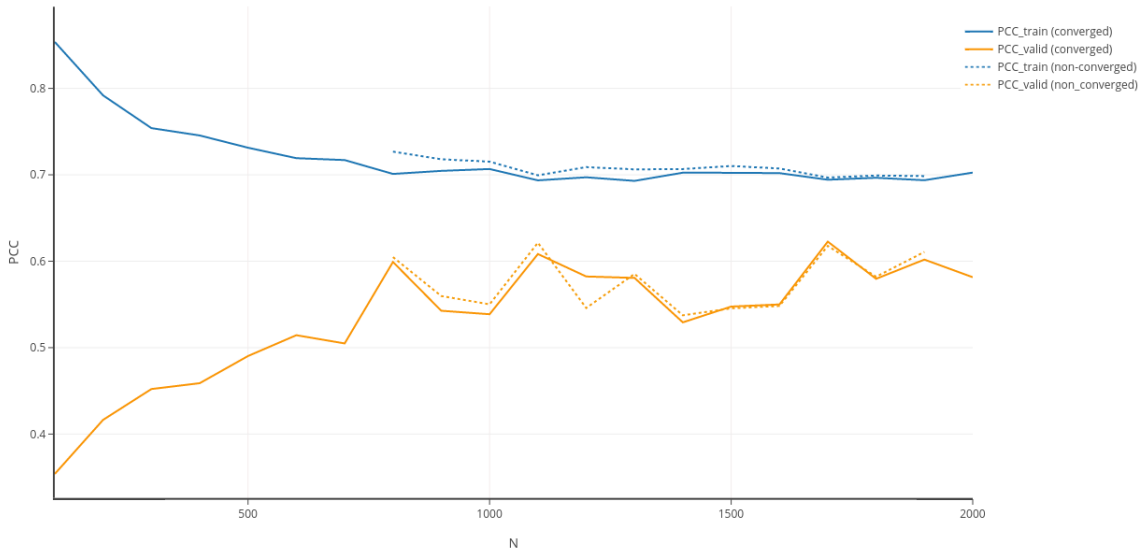


Figure 5.4: Performance in convergent versus non-convergent RVMs.

5.3.2 Predicting Binary Trait

Performance Measures

To assess predictive accuracy of the classification model, we used Sensitivity (SN), Specificity (SP) and Accuracy (ACC) measures.

Imbalanced Dataset

A dataset is imbalanced if the proportion of the classification categories are not approximately equal, such as our training dataset in which the proportion of the negative class to the positive class is almost $5 : 2$ for the binary trait. There are techniques to handle such datasets with SVMs such as assigning different weights

to positive and negative classes [10], called class-weight SVM. Experiments on our dataset indicate that the RVM is also sensitive to imbalanced data. Figure 5.5 shows an instance of the test results of three classifiers (class-weight SVM, ordinary SVM, and RVM) with linear kernel, trained with a stratified sampling of 800 training individuals. We can see that the best to the worst performances in order are the class-weight SVM (with 620 SVs), the RVM (with 38 RVs), and the ordinary SVM (with 523 SVs) classifiers.

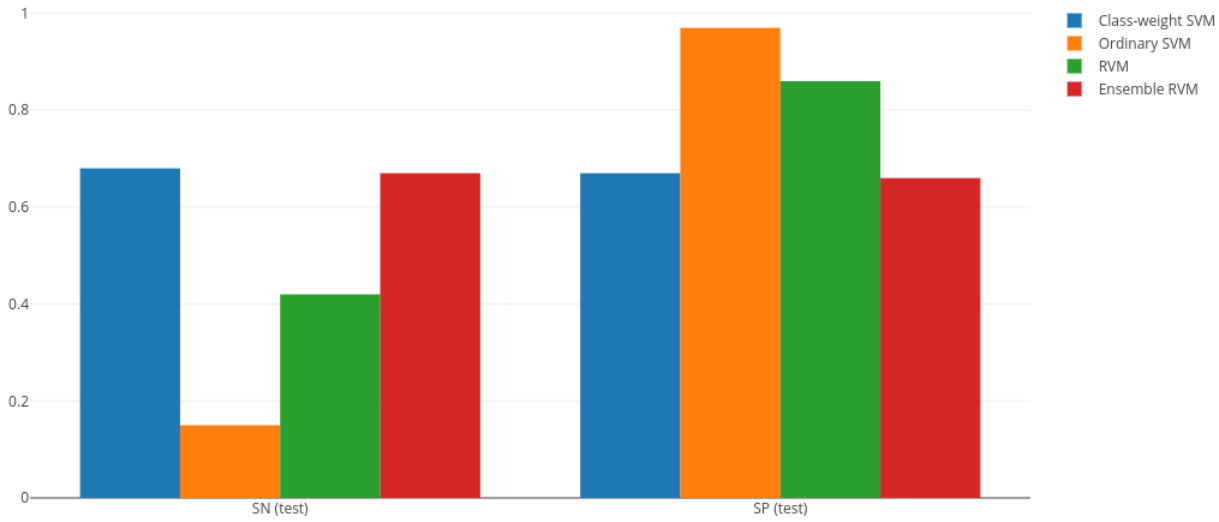


Figure 5.5: The sensitivity and specificity of the class-weight SVM, ordinary SVM, RVM, and Ensemble RVM classifiers over the test dataset.

Ensemble Architecture

One way of dealing with the problem of imbalanced datasets is employing ensemble predictors. Therefore, in a similar process to the Qtrait predictor, we constructed an ensemble with 100 RVMs and trained the ensemble using balanced bootstrap samples of 1300 items from the original training dataset. The SN, SP, and ACC of the trained ensemble on the test data are 0.67, 0.65, and 0.66, respectively, by taking majority vote. These results are now analogous to the class-weight SVM results, as shown in Figure 5.5.

About Relevance Vectors

There are about 1300 shared RVs in the ensemble. In the classification case, how the top ranked RVs represent prototypical examples might not be as clear as regression case. However, the imbalanced property of the dataset can be considered an attribute which we might expect to happen in the population of RVs, too. Interestingly, the shared RVs are also imbalanced, (i.e., for every five negative individuals that are in the top ranked RVs, there are about two positive individuals), even though each RVM in the ensemble was trained with a balanced sample.

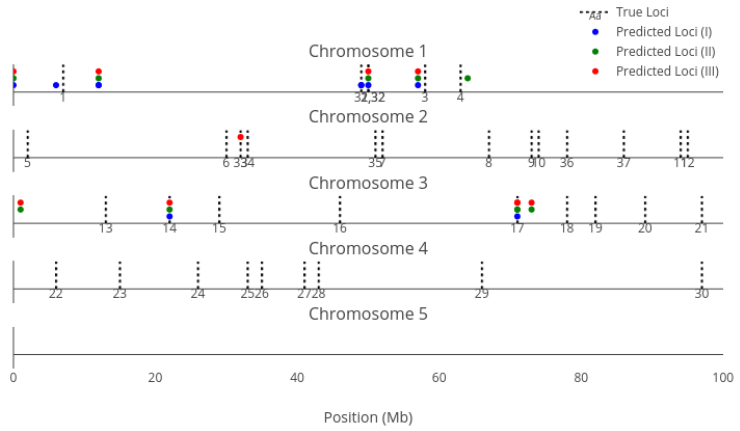
5.4 Identifying Influential Markers

We report the result according to three criteria [66]: the success rate (ratio of mapped QTL to the total number of simulated QTL), and the error rate (ratio of

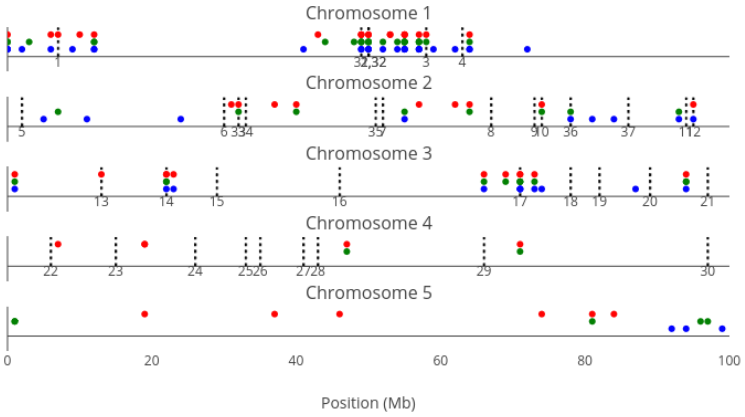
false positives to the number of reported positions), and mean distance between a true mapped QTL and the nearest reported positions. A true QTL is considered mapped if one or more of the predicted positions is within 1 Mb distance from the QTL. Predicted positions are considered as false positives, if a distance to the closest true QTL exceeded 1 Mb. It is possible that one predicted position is mapped to two different QTL, or two predicted positions they are considered to map the same true QTL.

5.4.1 Markers Affecting Quantitative Trait

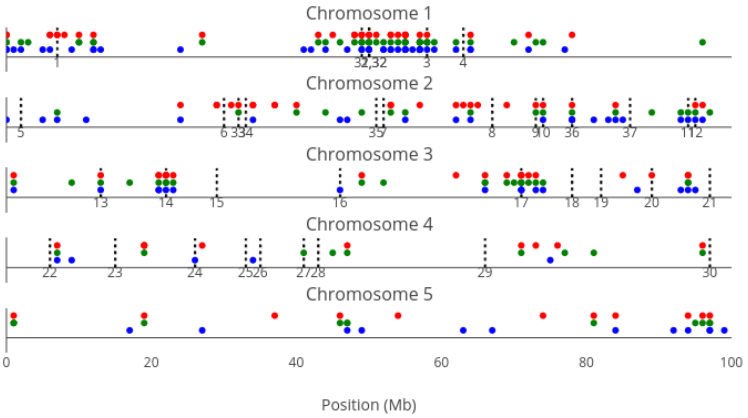
For ranking SNPs affecting the Qtrait, we used the RVs found by the ensemble Qtrait predictor to train three ensemble Qtrait SNP selectors: Ensemble (I) consists of 4000 RVMs trained with bootstrap samples of 50 individuals, Ensemble (II) consists of 2000 RVMs trained with bootstrap samples of 100 individuals, and Ensemble (III) consists of 1000 RVMs trained with bootstrap samples of 200 individuals. Figure 5.6 shows the true 37 Qtrait loci versus top ranked SNPs in either of three ensembles, if we consider the top ranked 10, 50 and 100 SNPs. For instance, Figure 5.6a shows that if we consider the top 10 SNPs, all the three ensembles identify the two major loci on Chromosome 3 (i.e., loci 14 and 17) plus one other additive loci on Chromosome 1 (i.e., 3) and one pair of epistatic on Chromosome 1 (i.e. loci 31 and 32).



(a)



(b)



(c)

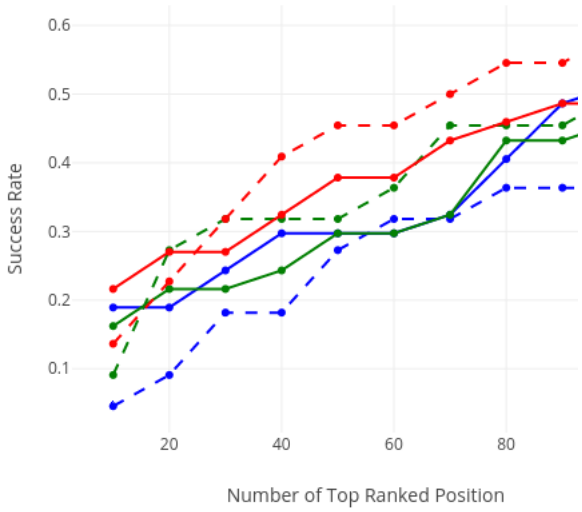
Figure 5.6: Predicted Important SNPs versus 37 QTL affecting Qtrait: (a) Top 10 Loci, (b) Top 50 Loci, and (c) Top 100 Loci. Dashed lines indicate to the true loci, while blue, green and red dots indicate to the positions of the identified important SNPs on each chromosome recognized by Ensemble RVM (I) ,(II) and(III), respectively. (Some loci, either true or predicted, overlapped due to their closeness).

5.4.2 Markers Affecting Binary Trait

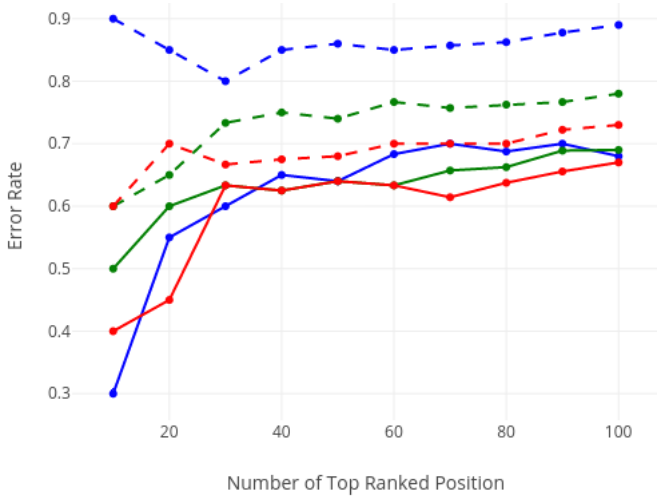
For ranking SNPs affecting the Btrait, we used the RVs found by the ensemble RVM Btrait predictor to train three ensemble RVM Btrait SNP selectors, all similar to the ensemble Qtrait SNP selectors (i.e., Ensembles (I)-(III)), but trained with balanced bootstraps. Figure 5.8 shows the true 22 Btrait loci versus top ranked SNPs in either of three ensembles, if we consider the top ranked 10, 50 and 100 SNPs. For instance, Figure 5.8a shows that if we consider the top 10 SNPs, ensemble (III) identifies three loci 1, 8 and 14, while ensemble (II) identifies two loci 1 and 14, and ensemble (I) identifies only locus 14.

5.4.3 Bootstrap Size versus Performance

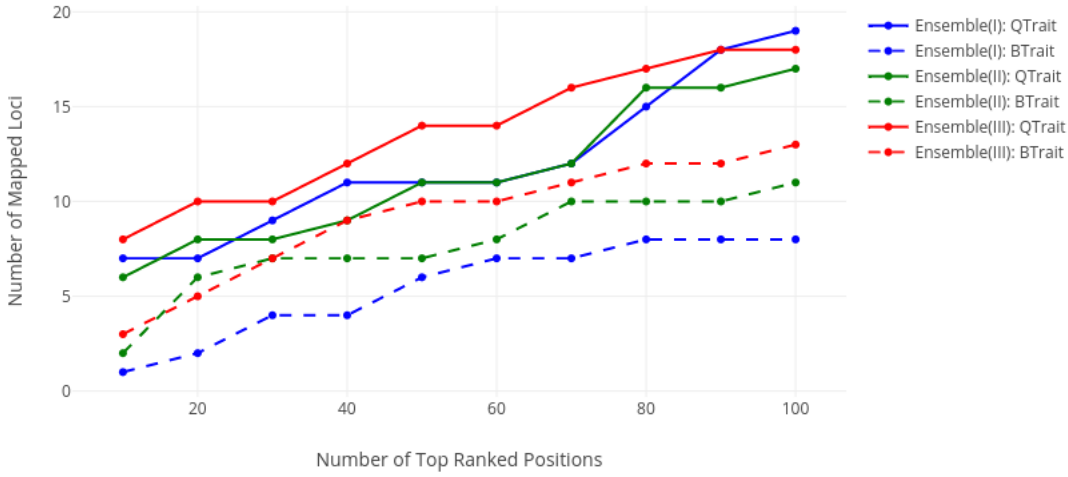
Figure 5.7 shows the performance of the three different ensemble architectures in identifying Qtrait and Btrait SNPs in terms of success rate, error rate and number of mapped loci in top ranked SNPs. These diagrams show that ensembles are clearly more successful in identifying influential Qtrait SNPs than Btrait SNPs. Also, among three ensemble architectures, ensemble (III) with a larger bootstrap sample size performs better (e.g., see 5.7c). However, different ensembles might be able to find different loci at different cut-off points of top ranked SNPs. For instance, as Figures 5.6c shows, ensemble (I) identifies loci 16 and 24, while these loci are not in top 100 SNPs in ensemble (II) and (III). Table 5.1 summarizes ensemble (III) performance in finding Qtrait and Btrait loci.



(a) Ensembles' success rate



(b) Ensembles' error rate



(c) Ensembles' mapped loci

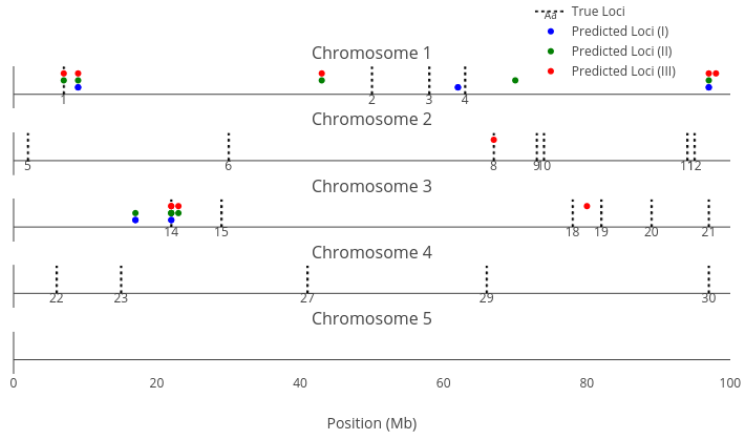
Figure 5.7: Ensembles' performance in identifying loci related to Qtrait and Btrait.

5.5 Comparison with Related Work

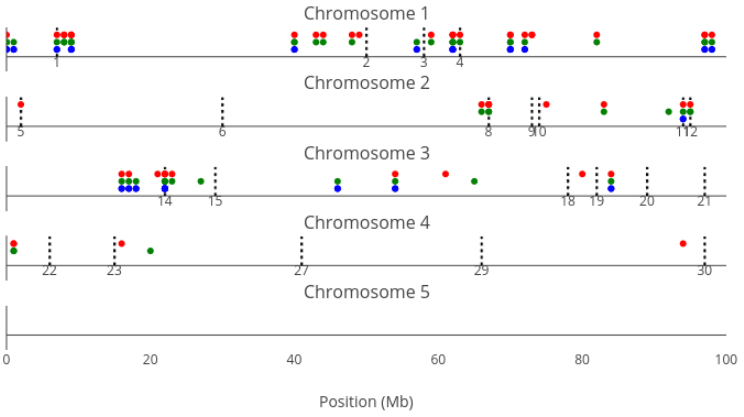
Previously, Ogutu et al. [69] compared SVMs, random forests and boosting for genomic selection over the same dataset. They use all the three methods for predict-

Table 5.1: Evaluating Ensemble (III) in identifying Qtrait and Btrait loci.

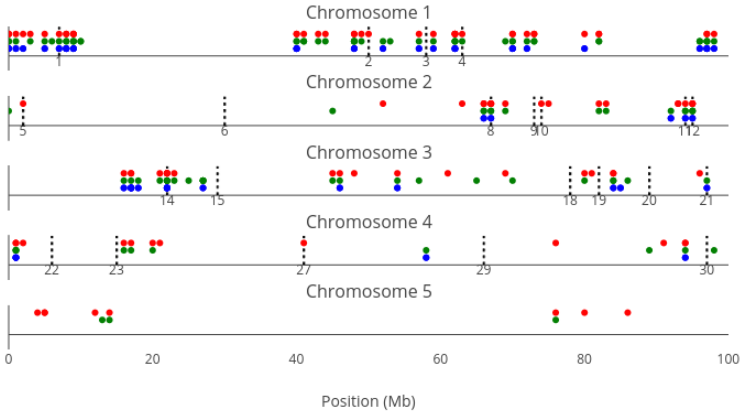
Feature type	Top positions	Mapped Loci	Mean dist. (Mb)	Success rate	Error rate
Qtrait loci	10	8	0.35	0.22	0.40
	100	18	0.48	0.48	0.67
Btrait loci	10	3	0.25	0.14	0.60
	100	13	0.40	0.59	0.73



(a)



(b)



(c)

Figure 5.8: Predicted Important SNPs versus 22 QTL affecting Btrait: (a) Top 10 Loci, (b) Top 50 Loci, and (c) Top 100 Loci. Dashed lines indicate to the true loci, while blue, green and red dots indicate to the positions of the identified important SNPs on each chromosome recognized by Ensemble RVM (I) ,(II) and(III), respectively. (Some loci, either true or predicted, overlapped due to their closeness).

ing Qtrait, and obtained the correlation coefficient values 0.547 for boosting, 0.497 for SVMs, and 0.483 for random forests. Comparing to their results, our ensemble RVM with PCC=0.505 outperforms SVM and random forests, but is outperformed by boosting. Ogutu et al. also employed random forests for importance rankings of the SNPs using two different measures of importance ranking (i.e., mean square error and node impurity). However, they confined the evaluation of the results to chromosome maps, and they did not report any other measure that we can compare our results with, precisely. Nevertheless, only by visually comparing the chromosome maps, obtained by random forest (Figure 5.9) and ensemble RVM (III) (Figure 5.6), we can see that the ensemble RVM approach outperforms the random forests, despite similarities. For example, the random forest apparently was not successful in identifying any locus on Chromosome 2, but the ensemble RVM ranked an epistatic locus on Chromosome 2 in top 10. We also can find similar examples in Chromosome 3. The random forest was not able to find any locus other than the two major loci, but ensemble RVM identifies 2 more loci in top 100 SNPs.

5.6 Conclusion

For the first time, we investigated how ensemble RVMs perform in predicting quantitative and binary traits (i.e., regression and classification), and identifying important markers affecting the traits in a GWAS with a simulated dataset [88]. We also illustrated some issues in applying RVMs along with approaches for dealing

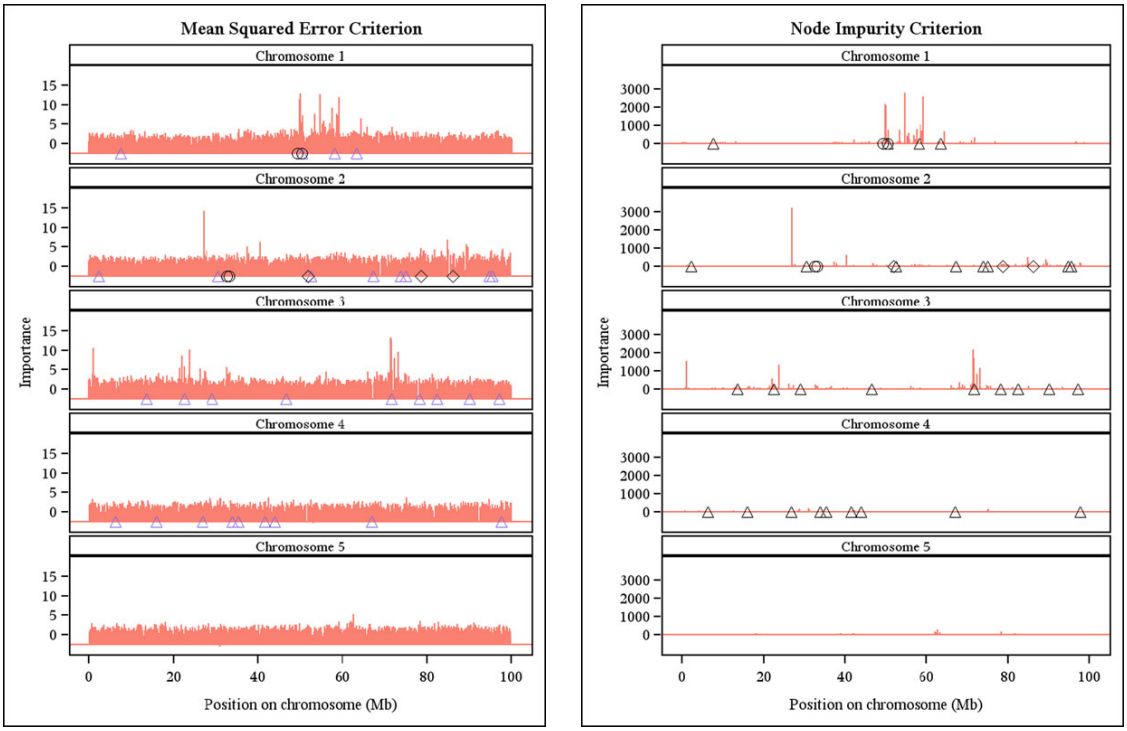


Figure 5.9: Importance ranking of the 10031 SNP markers by random forest. Positions of the simulated additive (triangle), epistatic (circle) and imprinted (diamond) QTLs are indicated on each chromosome. *Figure taken from Ogutu et al. [69], used under Creative Commons license.*

with them such as instability, non-convergency, and imbalanced datasets. RVMs (including ensemble RVMs) had been used neither for phenotype prediction nor for marker selection in genomic selection. We saw that RVMs can present positive results, thus at least they can be considered as an alternative approach to other strong machine learning methods such as SVMs in phenotype prediction.

The ensemble RVMs have similar advantages to a typical ensemble method, such as handling high-dimensionality problems, overcoming the imbalanced problems in classification cases, and improving the overall accuracy by decreasing model variance.

Apart from those advantages, our ensemble RVMs approach, either for phenotype prediction or for marker selection, have two main benefits, derived from the sparse solution property of RVMs. One advantage is that we can reduce a training dataset to its "representative" sequences (i.e., RVs), and even we can rank these representatives. In this way, we are able to recognize the sequences that are most important for regression or classification, so later a biologist might get more insights about the "best" and "worst" sequences. From this viewpoint, RVMs can be even used in a supervised clustering method. Another unique advantage of our ensemble RVMs is ability to rank the markers, and to identify the most important ones. Particularly in GWAS, when we seek a small set of important markers, ranking approach shows its usefulness.

Chapter 6

Experimental Results on Yeast

6.1 Introduction

In this chapter, we investigate how RVMs perform on genomic selection and GWAS with a real world and well-studied dataset such as yeast *Saccharomyces cerevisiae*. This yeast has been a popular model organism for biological research.

In genomic selection, we show how RVMs perform in predicting yeast growth in different environments. We will examine the effect of different kernels in our kernel RVMs. Also, we will demonstrate how ensembles of basis RVMs recognize and rank the most influential markers on the yeast growth in each environment for a GWAS.

6.2 Dataset

We used the yeast dataset which was originally from Bloom et al. [14]. They obtained this dataset from a study of 1,008 haploid yeast strains derived from a cross between a laboratory and a wine strain of the yeast *Saccharomyces cerevisiae*. The parent strains differed by 0.5% at the sequence level. The genotypes consist of markers that correspond to 11,623 sequence locations in the genome: coded as 1 if the sequence variation came from the wine strain parent; or 0 if it came from the laboratory strain parent.

Bloom et al. modified the environment of 1,008 yeast strains in 46 different ways (first column in Table 6.1), and measured the population growth under those different conditions. For example, they varied the basic chemicals used for growth (e.g. galactose, maltose), or added minerals (e.g. copper, magnesium chloride), then they measured growth trait in that condition. Precisely, Bloom et al. grew individuals for specific amount of time and took the image of the result colony, then they calculated the radius of the colony using image processing techniques. Some results, such as irregular colonies were removed and treated as missing data. Thus, there are less than 1,008 values for each trait. For example, there are only 599 reading available for Sorbitol. However, for most traits there are more than 900 readings.

6.3 Predicting Phenotypes

We considered the yeast dataset as 46 separate regression problems: we constructed a separate RVM model for predicting growth under each of 46 conditions. We trained each RVM with linear basis functions, linear kernels, Gaussian kernels (with different values of γ parameter), and a set of n-gram and compositional kernels. Using the coefficient of determination (R^2) as measure, and running 10 times of 10-fold cross-validation (each time with random different folds), we evaluated the results of RVM models. As the process for this dataset along with repeating cross-validations was computationally heavy and time-consuming, all the process has been done in parallel on the WestGrid (`www.westgrid.ca`) platform.

The accuracies plus the standard deviation of cross-validation accuracies in the best RVM model are shown in Table 6.1. Having a quick look at the results we can see that: (1) Gaussian kernel RVMs mostly produces promising results. Even in traits such as Mannose in which the linear kernel RVM shows a slightly better accuracy, it is possible to get the similar, if not better, accuracy with a bit more tuning of the Gaussian kernel parameter. The only exception is Cadmium Chloride in which linear Basis RVM presents a significantly better accuracy. (2) The RVM models are highly stable, as the standard deviations are small. In following subsections, we analyze the results with more details.

Table 6.1: Coefficient of determination (R^2) and standard deviation (std) of RVM predictions among the 46 traits. (Gaussian parameter: $\gamma_1 = 1e{-}4$, $\gamma_2 = 2e{-}4$, and $\gamma_3 = 3e{-}4$)

Trait	Linear Basis	Linear	Gaussian	10gram	10gram+pn	Best RVM	std
Cadmium Chloride	0.639	0.033	0.454	0.004	0.008	Linear Basis	0.005
Caffeine	0.074	0.216	0.233	0.01	0.018	Gaussian(γ_3)	0.006
Calcium Chloride	0.113	0.273	0.287	0.004	0.011	Gaussian(γ_3)	0.007
Cisplatin	0.133	0.29	0.287	0.011	0.016	Linear	0.006
Cobalt Chloride	0.258	0.439	0.466	0.005	0.016	Gaussian(γ_2)	0.006
Congo red	0.327	0.467	0.491	0.009	0.015	Gaussian(γ_1)	0.006
Copper	0.146	0.334	0.379	0.014	0.012	Gaussian(γ_3)	0.01
Cycloheximide	0.317	0.473	0.514	0.004	0.012	Gaussian(γ_1)	0.005
Diamide	0.277	0.473	0.483	0.014	0.012	Gaussian(γ_2)	0.005
E6 Berbamine	0.211	0.375	0.414	0.007	0.008	Gaussian(γ_2)	0.008
Ethanol	0.276	0.457	0.476	0.006	0.017	Gaussian(γ_2)	0.006
Formamide	0.114	0.207	0.25	0.007	0.015	Gaussian(γ_2)	0.006
Galactose	0.076	0.206	0.241	0.001	0.002	Gaussian(γ_3)	0.008
Hydrogen Peroxide	0.234	0.343	0.397	0.018	0.02	Gaussian(γ_2)	0.01
Hydroquinone	0.087	0.139	0.208	0.005	0.013	Gaussian(γ_3)	0.009
Hydroxyurea	0.12	0.296	0.342	0.01	0.015	Gaussian(γ_2)	0.01
Indoleacetic Acid	0.128	0.255	0.313	0.01	0.009	Gaussian(γ_2)	0.007
Lactate	0.36	0.542	0.555	0.011	0.02	Gaussian(γ_2)	0.005
Lactose	0.374	0.553	0.574	0.007	0.014	Gaussian(γ_2)	0.008
Lithium Chloride	0.531	0.597	0.678	0	0.006	Gaussian(γ_1)	0.006
Magnesium Chloride	0.102	0.245	0.255	0.003	0.015	Gaussian(γ_3)	0.005
Magnesium Sulfate	0.187	0.366	0.41	0.005	0.015	Gaussian(γ_3)	0.005
Maltose	0.409	0.484	0.523	0.005	0.011	Gaussian(γ_2)	0.005
Mannose	0.079	0.213	0.197	0.006	0.01	Linear	0.007
Menadione	0.216	0.389	0.411	0.011	0.015	Gaussian(γ_3)	0.006
Neomycin	0.422	0.583	0.596	0.005	0.021	Gaussian(γ_2)	0.003
Paraquat	0.31	0.442	0.454	0.012	0.017	Gaussian(γ_2)	0.005
Raffinose	0.185	0.385	0.388	0.013	0.023	Gaussian(γ_3)	0.007
SDS	0.199	0.36	0.398	0.007	0.014	Gaussian(γ_2)	0.004
Sorbitol	0.176	0.343	0.364	0.017	0.022	Gaussian(γ_3)	0.009
Trehalose	0.326	0.48	0.503	0.014	0.022	Gaussian(γ_2)	0.005
Tunicamycin	0.417	0.594	0.622	0.007	0.013	Gaussian(γ_1)	0.006
4-Hydroxybenzaldehyde	0.23	0.34	0.367	0.016	0.011	Gaussian(γ_2)	0.008
4NQO	0.44	0.496	0.512	0.005	0.018	Gaussian(γ_2)	0.005
5-Fluorocytosine	0.215	0.323	0.378	0.015	0.015	Gaussian(γ_2)	0.008
5-Fluorouracil	0.326	0.505	0.559	0	0.008	Gaussian(γ_2)	0.005
6-Azauracil	0.152	0.3	0.304	0	0.014	Gaussian(γ_3)	0.005
Xylose	0.282	0.455	0.478	0.005	0.012	Gaussian(γ_3)	0.004
YNB	0.379	0.224	0.515	0.001	0.002	Gaussian(γ_1)	0.009
YNB:ph3	0.059	0.18	0.177	0.002	0.014	Gaussian(γ_3)	0.005
YNB:ph8	0.203	0.327	0.361	0.008	0.016	Gaussian(γ_2)	0.006
YPD	0.368	0.266	0.511	0	0.001	Gaussian(γ_1)	0.008
YPD:15C	0.211	0.334	0.356	0.002	0.013	Gaussian(γ_2)	0.006
YPD:37C	0.473	0.566	0.611	0.007	0.013	Gaussian(γ_2)	0.006
YPD:4C	0.18	0.406	0.438	0.016	0.014	Gaussian(γ_2)	0.005
Zeocin	0.316	0.46	0.475	0.01	0.018	Gaussian(γ_3)	0.004

6.3.1 Linear Kernel RVM versus Linear Basis RVM

We presented definitions for linear RVMs in the form of linear kernel and linear basis RVMs in Section 3.1. We explained that a linear basis RVM can be viewed as an RVM with no basis function, as we use input vectors directly in the data model instead. Similarly when we use linear kernels, in fact we have no kernel. It means there is no feature space, so our estimator tries to pass a hyperplane through input vectors in the input space (e.g., in regression case). Here, we might expect that both linear kernel and linear basis RVMs produce similar results or with subtle difference, as both are linear and in the same space. However, that is not the case, i.e., linear kernel RVM and linear basis RVM produces different hyperplanes as we see in the results in Table 6.1. Consider Cadmium Chloride and YPD:4C, as two extreme examples. In the former, linear basis RVM has high accuracy, while in the latter linear kernel RVM shows higher accuracy. As a corollary we can say that linear basis RVM produces results which classic linear SVM is not able to. We know that the linear kernel cannot be more accurate than a properly tuned Gaussian kernel [49], but we cannot conclude the same for the linear basis function. Therefore, even if we have conducted a complete model selection using the Gaussian kernel RVM for a problem, it is still valuable to consider the linear basis RVM, just as we saw linear basis superiority to Gaussian kernel in Cadmium Chloride.

6.3.2 Investigating String Kernel RVM

We have also examined a set of string kernels with RVM: several n-gram kernels ($n = 3, 5, 7, 10$) alone or by composing with polynomial kernels. As samples, we have shown the result of 10-gram kernel alone and its composition with a polynomial kernel of degree 100 (i.e., $(\alpha \times k_n(x, y) + \beta)^{100}$) in Table 6.1. All string kernels showed poor accuracies on our dataset. The issue arises from the fact that a typical n-gram kernel on this dataset gives us a Gram matrix with almost all elements close to one. For instance, if we round the 3-gram kernels to their nearest thousandth, we will have a matrix of ones. It intuitively indicates that the sequences are so similar to each other that the predictor cannot discriminate between any pairs. Although if we increase n or compose the n-gram with other kernels, such as polynomial, we may see improvement in the results (e.g., compare the column 10-gram+pn to the column 10-gram in Table 6.1), this improvement is insignificant.

We think genetic linkage can be a reason for n-gram kernels adversity in this problem. Genetic linkage describes an inheritance tendency in which two markers located in close proximity to each other on the same chromosome are inherited together during meiosis[62]; i.e, the nearer two genes are on a chromosome, the lower the chance of recombination between them, and the more likely they are to be inherited together. On the other hand, n-gram kernels capture the short adjacent similarities in sequences. Therefore, high similarity between sequences captured by n-gram kernels comes as no surprise. That is, we expect the small 3-10 SNP se-

quences to be shared between individuals because these sequences appear close to each other in the genome and are similar due to genetic linkage. The genetic linkage phenomenon can also illustrate why n-gram kernels previously helped for gene-scale problems such as metabolic network prediction [74], but do not work for this problem which has a genome-scale attribute.

6.3.3 Heritability versus Accuracies

Bloom et al. [14] provided estimates for narrow-sense and broad-sense heritability for the yeast dataset. They considered broad-sense heritability as the contribution of additive genetic factors (i.e., narrow-sense heritability) and gene-gene interactions. Accordingly, the difference between the two heritability measures provides an estimate of the contribution of gene-gene interactions. Among the 46 traits, broad-sense heritability estimates ranged from 0.40 (YNB:ph3) to 0.96 (Cadmium Chloride), with a median of 0.77. Narrow-sense heritability estimates ranged from 0.21 (YNB:ph3) to 0.84 (Cadmium Chloride), with a median of 0.52. Using the difference between two heritability measures, we can estimate the fraction of genetic variance due to gene-gene interactions, which ranged from 0.02 (5-Fluorouracil) to 0.54 (Magnesium Sulfate), with a median of 0.30. Therefore, the genetic basis for variation in some traits, such as 5-Fluorouracil, is almost entirely due to additive effects, while for some others, such as Magnesium Sulfate, approximately half of the heritable component is due to gene-gene interactions.

To see if there is a correlation between heritability and RVM prediction accuracies, we calculated the PCC between estimates of heritability and prediction accuracies. The correlation coefficients in three RVM categories (Gaussian, linear, and linear basis) have been shown in Figure 6.1. The values related to the broad- and narrow-sense heritability (blue and orange bars) indicate that heritability and RVM accuracies, particularly in Gaussian and linear basis RVMs, have strong positive association. In other words, we will have better predictions when the amount of heritability increase. Especially if there is a higher narrow-sense heritability (more additive effects), we can expect better results from the RVM predictor.

Does this association imply that RVM will be less successful in predicting traits in which the genetic variation is more related to gene-gene interactions? To respond to this question, we also calculated the correlation coefficient between RVM accuracies and gene-gene interactions effects (green bars in the figure). These values indicate that gene-gene effects and accuracies, particularly in Gaussian and linear RVMs, have small negative association, which means we cannot infer the RVM performance is deteriorating when gene-gene interactions effects increases. We might have expected that result, as an RVM model is capable of taking account all sorts of factors for prediction. However, if we have narrow-sense heritability estimates before constructing an RVM model, we are able to anticipate behaviour of the predictor, due to the higher weight of additive effects (as most genetic variance in populations is additive [32]).

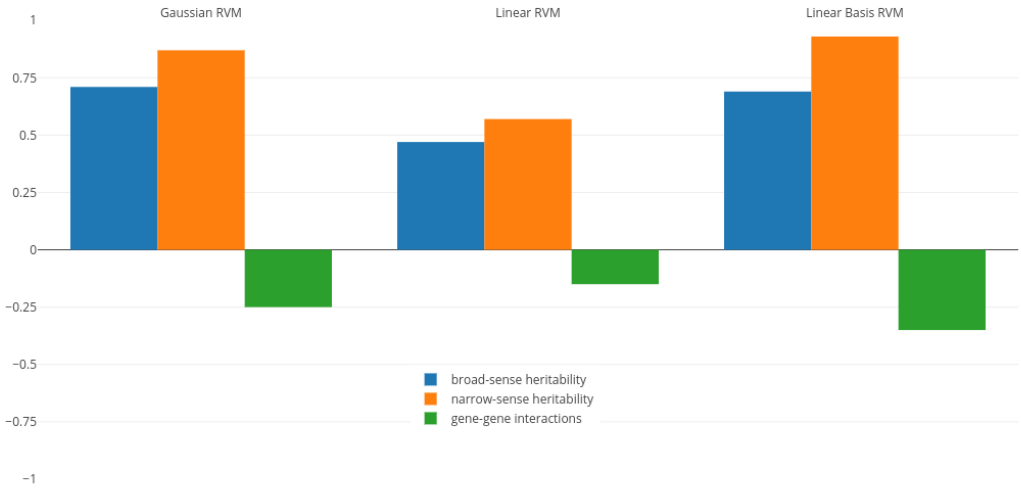


Figure 6.1: Pearson correlation coefficient between RVM accuracies and different heritability measures.

6.3.4 Comparison with Related Work

Grinberg and King [38] recently compared several learning methods including forward stepwise regression, ridge regression, lasso regression, random forest, gradient boosting machines (GBM), and Gaussian kernel SVM with two classical statistical genetics methods (BLUP and a linkage analysis done by Bloom et al. [14]). Grinberg and King used the coefficient of determination (R^2) as the accuracy measure, and evaluated their models with one run of 10-fold cross validation. In Table 6.2, the columns "GK: Best of Others" and "GK: SVM" refer to Grinberg and King's results.

Comparing to the SVM, RVM models show better predictions, overall. However, Grinberg and King's approach for training and model selection in Gaussian SVM

does not seem proper. Grinberg and King trained an SVM for each fold of cross-validation. In other words, they trained 10 SVMs (10 sets of Gaussian kernel and SVM parameters) for a trait. In this way, not only the accuracies are overestimated, but also the model selection process appears problematic (e.g., the set of parameters that should be used to predict a trait for new yeast individuals is unclear). Nevertheless, RVM shows superiority despite overestimated SVM accuracies.

Comparing to the best of the other methods, RVM turned out to have more or less identical performance with others, except in six traits including Cadmium Chloride, Indoleacetic Acid, Magnesium Sulfate, Maltose, 4NQO, and YPD:37C in which GBM or Bloom et al.'s method showed superiority. However, we should note that we do not know about the stability of the methods experimented by Grinberg and King, as they run only one 10-fold cross- validation. On the other hand, RVM shows high stability, as its standard deviations in 10 runs of 10-fold cross-validation were small.

6.4 Identifying Influential Markers

For identifying the most influential markers (SNPs) on the traits, we used our RVM ensemble architecture for ranking markers. An ensemble for a trait was composed of 400 linear basis RVMs, each with subsampling 50 to 60% of training data. As we are only interested in a small set of top ranked markers, the size of subsampling does not make a difference in the results (data not shown). To demonstrate how

Table 6.2: Our RVM results versus Grinberg and King's (GK) [38]. For the RVM column, the R^2 value belongs to the best RVM given in Table 6.1.

Trait	GK: Best of Others	GK: SVM	RVM(std)
Cadmium Chloride	GBM:0.797	0.565	0.639(0.005)
Caffeine	GBM:0.250	0.234	0.233(0.006)
Calcium Chloride	BLUP: 0.268	0.261	0.287(0.007)
Cisplatin	GBM: 0.338	0.272	0.287(0.006)
Cobalt Chloride	GBM: 0.460	0.448	0.466(0.006)
Congo red	Lasso:0.504	0.487	0.491(0.006)
Copper	GBM:0.452	0.338	0.379(0.01)
Cycloheximide	SVM:0.529	0.529	0.514(0.005)
Diamide	BLUP:0.498	0.486	0.483(0.005)
E6 Berbamine	GBM:0.412	0.390	0.414(0.008)
Ethanol	GBM:0.518	0.455	0.476(0.006)
Formamide	GBM:0.350	0.240	0.25(0.006)
Galactose	GBM:0.235	0.217	0.241(0.008)
Hydrogen Peroxide	SVM:0.399	0.399	0.397(0.01)
Hydroquinone	BLUP:0.225	0.188	0.208(0.009)
Hydroxyurea	GBM:0.337	0.301	0.342(0.01)
Indoleacetic Acid	Bloom et al.:0.480	0.3	0.313(0.007)
Lactate	Lasso:0.568	0.557	0.555(0.005)
Lactose	GBM:0.582	0.565	0.574(0.008)
Lithium Chloride	GBM:0.711	0.680	0.678(0.006)
Magnesium Chloride	Bloom et al.:0.278	0.267	0.255(0.005)
Magnesium Sulfate	Bloom et al.:0.519	0.378	0.41(0.005)
Maltose	GBM:0.809	0.522	0.523(0.005)
Mannose	GBM:0.255	0.215	0.213(0.007)
Menadione	GBM:0.432	0.402	0.411(0.006)
Neomycin	Lasso:0.614	0.597	0.596(0.003)
Paraquat	Lasso:0.496	0.479	0.454(0.005)
Raffinose	GBM:0.383	0.364	0.388(0.007)
SDS	Lasso:0.411	0.383	0.398(0.004)
Sorbitol	Bloom et al.:0.424	0.318	0.364(0.009)
Trehalose	GBM:0.515	0.477	0.503(0.005)
Tunicamycin	SVM:0.634	0.634	0.622(0.006)
4-Hydroxybenzaldehyde	GBM:0.397	0.36	0.367(0.008)
4NQO	GBM:0.636	0.542	0.512(0.005)
5-Fluorocytosine	GBM:0.399	0.364	0.378(0.008)
5-Fluorouracil	Lasso:0.552	0.546	0.559(0.005)
6-Azauracil	GBM:0.315	0.279	0.304(0.005)
Xylose	GBM:0.516	0.460	0.477(0.004)
YNB	GBM:0.543	0.525	0.515(0.009)
YNB:ph3	BLUP:0.195	0.166	0.177(0.005)
YNB:ph8	BLUP:0.356	0.334	0.361(0.006)
YPD	GBM:0.556	0.524	0.511(0.008)
YPD:15C	Bloom et al.:0.432	0.333	0.356(0.006)
YPD:37C	Bloom et al.:0.711	0.603	0.611(0.006)
YPD:4C	GBM:0.485	0.421	0.438(0.005)
Zeocin	GBM:0.495	0.475	0.472(0.004)

well the ensemble RVMs act in identifying influential markers, we present the top ranked markers in three traits: Cadmium Chloride, Lithium Chloride, and Mannose. We chose Cadmium Chloride and Mannose as samples which the linear basis RVM showed excellent and poor phenotypic accuracies (Table 6.1), respectively, while we chose Lithium Chloride for comparison purposes that are relevant in the next section.

The ensemble RVMs in each of the three traits ranked around 90% of the markers with rank values in the range $[1, 400]$. The unranked markers indicate the markers that do not have any effect (even minor) on a trait. The top ranked markers can be chosen based on a threshold. For example, we can define the most influential markers as those who are chosen by half of the RVMs in the ensemble as RVs, so in this dataset we will have less than ten influential markers in the three traits. The ranked markers indicate those who may have positive or negative effects on a trait. In other words, we not only find the markers which have additive effects on yeast growth in an environment such as Lithium Chloride, but also we find those which have adverse effects on growth.

6.4.1 Comparison with Related Work

Previously, Bloom et al. [14] conducted a linkage analysis in the yeast cross (same dataset) with high statistical power to map functional QTL in all 46 traits. They found that nearly the entire additive genetic contribution to heritable variation (narrow-sense heritability) in yeast can be explained by the detected loci. Bloom

et al. specifically showed that for one trait (Lithium Chloride), the loci detected by their method explained most of the heritability. Nevertheless, it is still important to check the prediction performance in a validation population [40, 80]. Also, biological information or experiments might be required to confirm the result [80].

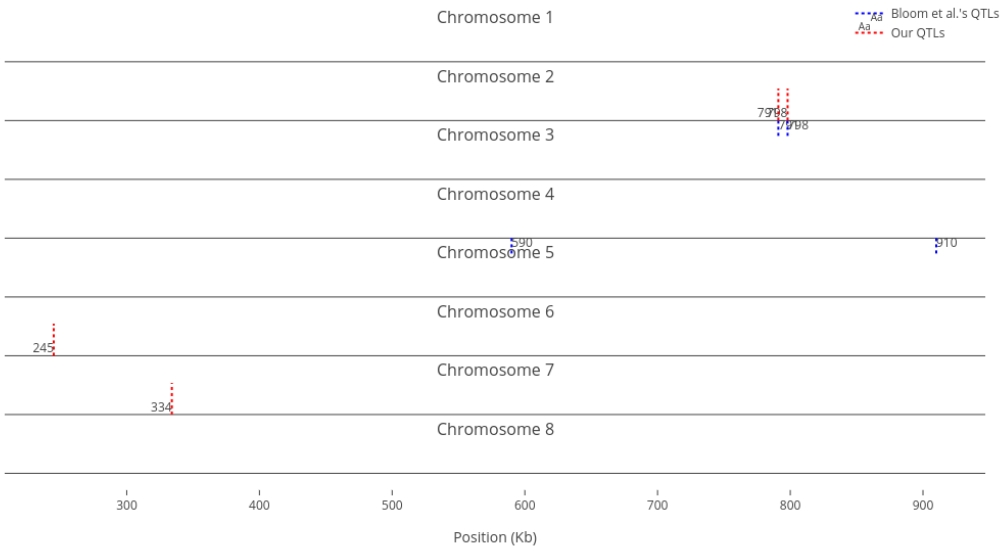
Bloom et al.'s study was the only research on yeast, at the time of writing, that had identified QTL (though only additive). Also, recent research refers to their findings (e.g, [32]). Thus, to see how well our RVM ensembles results are, we compare our identified influential markers in three traits to Bloom et al.'s QTL. Bloom et al. found 6, 22, and 10 additive QTL in Cadmium Chloride, Lithium Chloride, and Mannose, respectively. Therefore, we chose the top 6, 22, 10 ranked SNPs in the three traits as well. Figures 6.2, 6.4, and 6.5 show both results in each of the three traits accordingly. Each of the figures includes two parts (a) and (b) corresponding to the map of yeast chromosomes 1-8 and 9-16, respectively. The results were very promising: the markers identified by the RVM ensembles have similar distribution to the Bloom et al.'s QTL. Also, the RVM ensembles were relatively successful in finding the exact markers in the traits (33% match rate in Cadmium Chloride, 36% in Lithium Chloride, and 40% in Mannose). It is also interesting that the highest match rate among the three traits belongs to Mannose in which the linear basis RVM had poor prediction accuracy. This could be an indication of an RVM being capable of recognizing true "representatives" of a population, despite unacceptable predictions. Another advantage is in the ranking system with which we can always

recognize the most to the least markers' weight on a trait (even in the small set of top ranked sites). However, we can also go further and conclude that those top ranked markers who are close to each other (e.g, markers at loci 649 kb, 656 kb, and 677 kb on Chromosome 12 in Figure 6.4) might imply to the higher impact of a locus near to those markers on a trait due to genetic linkage.

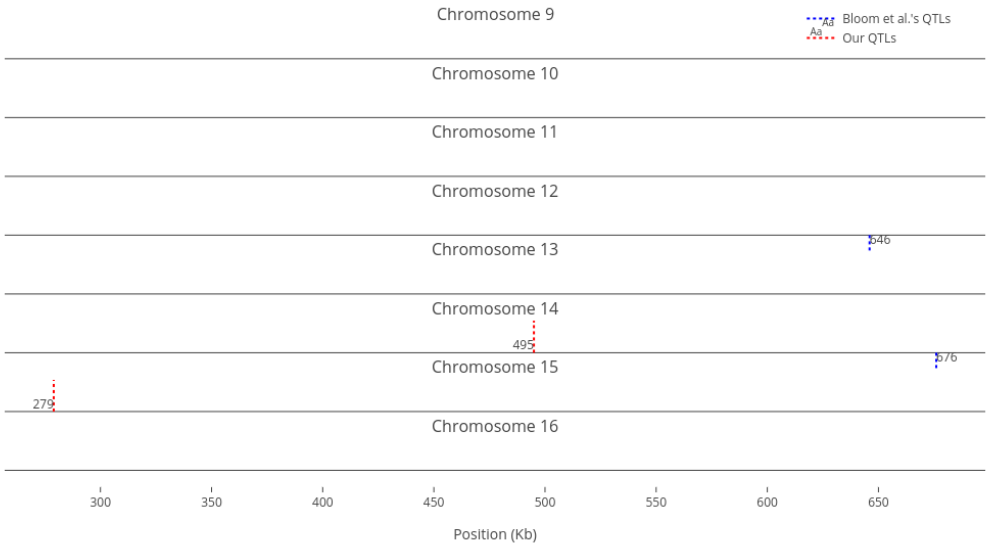
For comparison purposes, we only demonstrated an equal number of top ranked markers to Bloom et al.'s QTL. However, if we decreased the threshold, the number of influential markers would increase, so we might have witnessed a higher match rate. For instance, Figure 6.3 shows the top ten (instead of six) most influential markers in Cadmium Chloride. In this case, another additive QTL in chromosome 12 is identified (i.e., at position 464 kb). Another point that we should note is that not all influential markers on a trait have additive effects. Therefore, the identified markers which are distant from Bloom et al.'s QTL, present a good set of candidates for further investigation by a biologist, to see if they have non-additive effects with other loci. Also, as Bloom et al.'s results are not verified, the results that we show may be recognizing additional additive markers not located by previous results.

6.5 Conclusion

In this chapter, we studied how RVM performs on growth prediction of yeast in 46 different environments, comparing its performance with some other learning methods such as SVM and GBM. Our obtained phenotype prediction accuracies

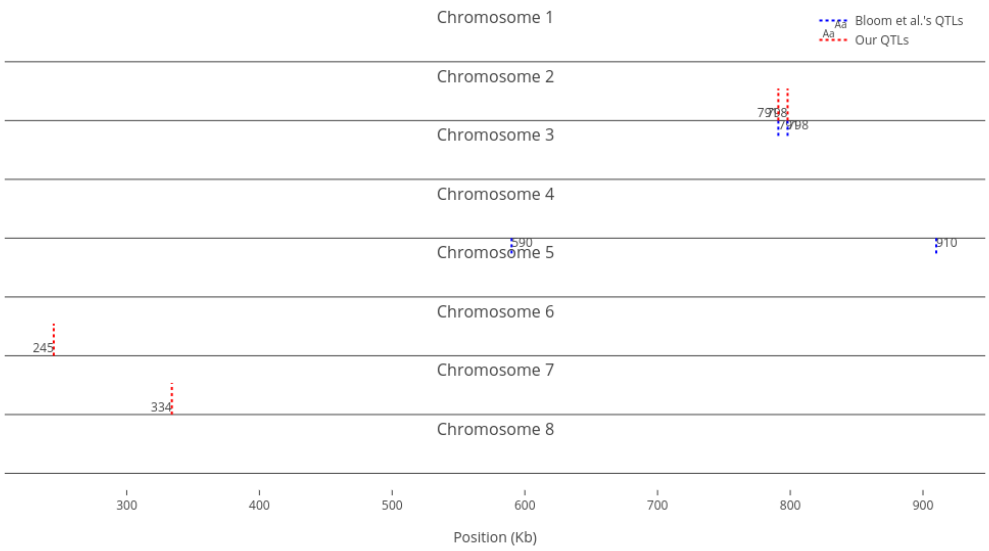


(a)

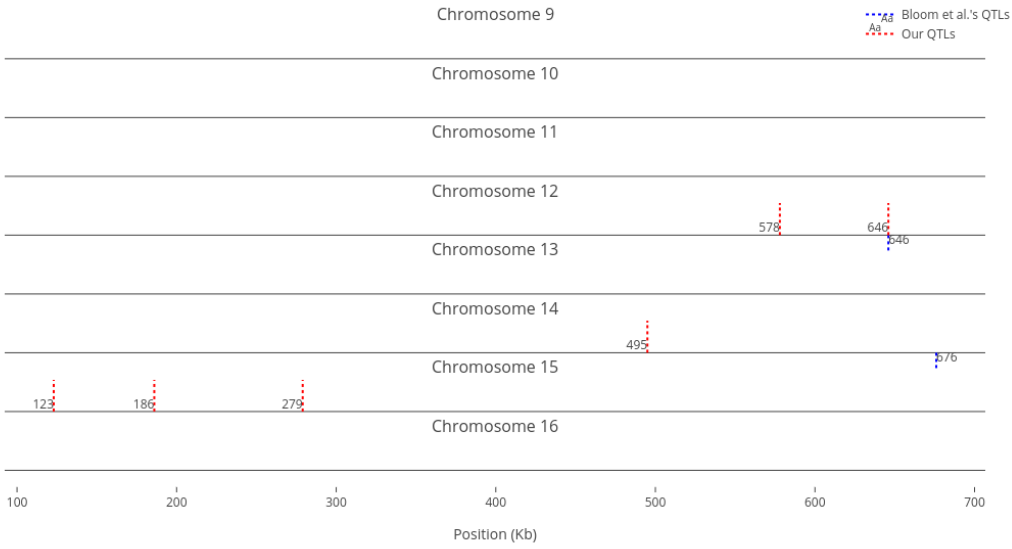


(b)

Figure 6.2: Top 6 influential markers on growth in Cadmium Chloride recognized by ensemble RVMs versus Bloom et al.'s 6 QTL.

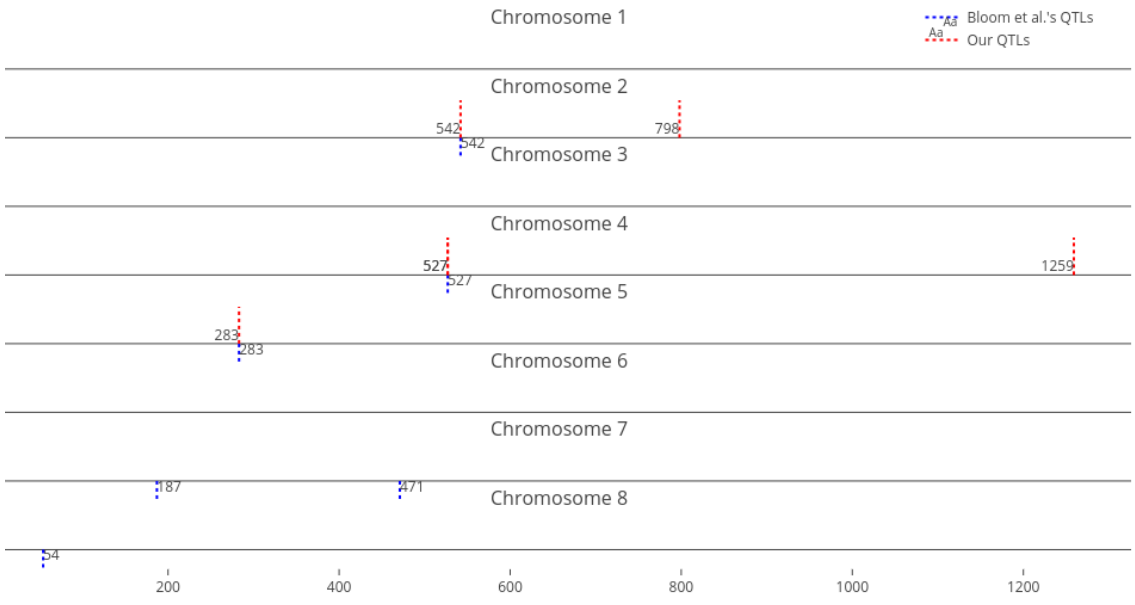


(a)

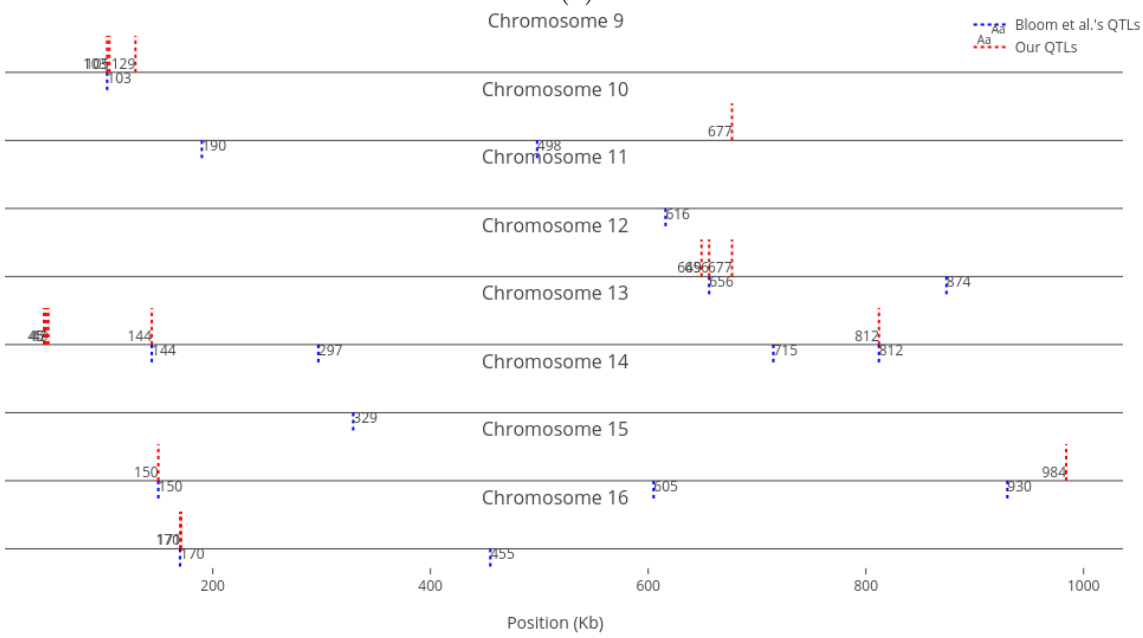


(b)

Figure 6.3: Top 10 influential markers on growth in Cadmium Chloride recognized by ensemble RVMs versus Bloom et al.'s 6 QTL.

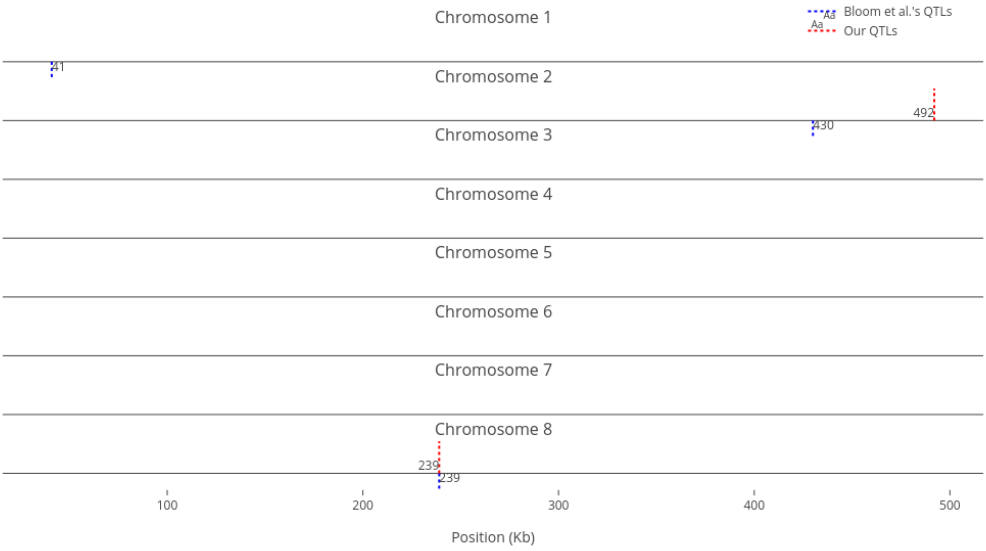


(a)

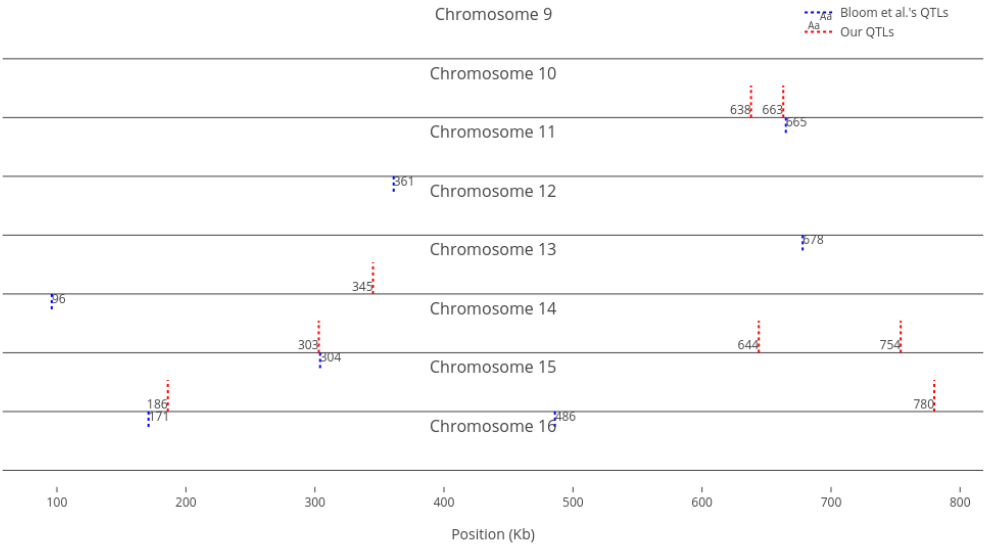


(b)

Figure 6.4: Top 22 influential markers on growth in Lithium Chloride recognized by ensemble RVMs versus Bloom et al.'s 22 QTL. (To view the improved version of (b) follow the link:(b) - Interactive Chart)



(a)



(b)

Figure 6.5: Top 10 influential markers on growth in Mannose recognized by ensemble RVMs versus Bloom et al.'s 10 QTL.

suggest that RVM shows positive results, and can be used as an alternative method (i.e., a method which is as valid as other successful learning methods) in genomic selection. Our intention, however, was not to show the superiority of RVM, as each machine learning method has its own pros and cons.

We investigated different kernels in RVM. We illustrated how different linear RVMs, i.e, linear kernel RVM and linear basis RVM perform in phenotype prediction. We observed that Gaussian RVMs had the best accuracies, while string kernel RVM, such as n-gram, presented poor predictions. We think this poor performance is due to the genetic linkage phenomenon. To our knowledge, this research is the only study that has been used string kernels in genomic selection, so analysing these kernels will open the door for future research.

We also investigated the relationship between different heritability measures and RVM prediction accuracies. The results indicate an strong association between narrow-sense heritability and prediction accuracy in RVMs. On the other hand, new research points out that the most genetic variance in populations is additive [32]. Knowing these facts, we can consequently anticipate the performance of the model before constructing it.

The last part of this chapter was devoted to identifying most influential markers on the traits. We chose three traits with different phenotype prediction accuracies as samples, and demonstrated how well our RVM ensembles work to rank the markers in each trait, comparing the results with other research which used a traditional

linkage analysis to find additive QTL. The comparison validated the results of RVM ensembles in finding markers with additive effects. However, we can earn more from the RVM ensembles, as those are capable of identifying both growth-increasing and growth-decreasing markers in yeast.

Chapter 7

Experimental Results on Flax

7.1 Introduction

In this chapter, we investigate how RVMs perform on genomic selection and GWAS with flax (linseed, or *Linum usitatissimum L.*) data. Flax is a crop that is commercially grown as a source of stem fibre and seed oil. Flax seeds contain oil composed of main fatty acids such as linoleic (LIO) and linolenic (LIN) acids. The major breeding aims of oilseed flax development are high seed yield, high oil content, and different levels of LIN contents (high or low) [104]. High-LIN flaxseed is one of the richest dietary sources of omega-3 fatty acid, whereas low-LIN flaxseed improves the oxidative stability and suitability of linseed oil for food products [30]. The major breeding aims of fibre flax are increased straw yield, fibre content in straw, fibre quality, and resistance to disease and abiotic stresses [104]. Flax straw

and its processed forms are mainly used in the pulp and paper industry. Other uses of flax straw such as producing bio-based materials for industrial applications are being researched [31]. Canada is the world's leader in the production and export of flax [29].

In genomic selection, we show how RVMs perform in predicting seed yield, oil content, iodine value, linoleic, and linolenic acid content in three different populations of flax. We will examine the effect of different kernels in our kernel RVMs. Also, we will demonstrate how ensembles of basis RVMs recognize and rank the most influential markers on the aforementioned traits for a GWAS on flax.

7.2 Dataset

We used the flax dataset of You et al. [104]. This dataset contains three bi-parental flax populations:

1. BM was generated from a cross between two high-yielding, moderately high-LIN, Canadian linseed parents. This population consists of 243 lines, each with 342 markers.

2. EV was generated from a cross between a low-LIN breeding line and a fibre flax cultivar. It consists of 86 lines, each with 443 markers.

3. SU was obtained from a cross between a low-LIN and high-LIN breeding lines. It comprised of 70 lines, each with 474 markers.

There are about 100 markers shared between these populations. Genetic Markers in this flax dataset are Single Sequence Repeat (SSR) markers. SSRs are stretches of DNA consisting of variable number of short tandem repeats. These SSRs are different between even closely related species. As an SSR in our dataset has either of two values, we can represent the individuals as sequences of 0 and 1. The three full marker sets cover about 74% of the flax genome, showing that the markers have been distributed genome-wide in the populations [104], though not dense, like the yeast dataset in Chapter 7.

There were some missing genotype data in the populations: 16% in BM, 1.4% in EV, and 4.5% in SU. For each population, we used most frequent strategy along a marker axis for data imputation. That is, if a marker X was missing in a sample, then the most frequent value among all samples was used for marker X. We considered values of five traits for these populations as phenotypic data: iodine value (IOD), linoleic acid content (LIO), linolenic acid content (LIN), oil content (OIL), and seed yield (YLD). All of these traits are quantitative. Based on the broad-sense heritability estimations done by You et al. over the two populations EV and SU, the traits LIN, LIO, and IOD have high, OIL has intermediate, and YLD has low heritability.

7.3 Predicting Phenotypes

For every pair of trait and population, we constructed a regression RVM (i.e., 15 models in total). We trained each RVM with following functions: linear basis, linear kernel, Gaussian kernel (with different values of γ parameter), and a set of n-gram kernels (5-gram to 20-gram). The result of 10 times of 5-fold cross-validation (each time with different random folds) has been shown in Table 7.1. We measured the accuracy of the genomic predictions using the Pearson's correlation coefficient (PCC) between the predicted and observed phenotypic values. Also, we calculated the standard deviation of the PCC for each model.

These results indicate that excluding the n-gram RVM, the other three RVMs do acceptable predictions, where the Gaussian and linear basis RVMs provide better results compared to the linear kernel RVM. Similar to the yeast dataset 7, N-gram kernels does not predict well here. This is another evidence in support of non-suitability of N-gram kernels in genomic selection due to genetic linkage; however, the Gram matrices here did not have severe "matrix of ones" issue which we had in yeast (Section 6.3.2). The best predictions among n-gram kernels belong to (BM, LIO) and (SU, LIO) using 9-gram kernel and 5-gram kernel, respectively.

In general, we have higher accuracies in EV compared to BM and SU. This superiority indicates that the markers specific in the EV population have more importance on the traits than those in the BM or SU, in combination with the common set markers. Also, the results show that the phenotype prediction accuracies and heritability

Table 7.1: Pearson correlation coefficient and standard deviation of predictions among the five traits in three populations.

Population	Trait	Linear Basis	Linear	Gaussian	N-gram	You et al. [104]	Best
BM	IOD	0.12(0.05)	-0.03(0.02)	0.16(0.03)	-0.05(0.01)	0.28(0.14)	You et al.
	LIO	0.45(0.03)	0.27(0.03)	0.61(0.01)	0.32(0.01)	0.36(0.11)	Gaussian
	LIN	0.28(0.04)	0.06(0.02)	0.34(0.03)	-0.03(0.01)	0.43(0.12)	You et al.
	OIL	0.32(0.03)	0.06(0.02)	0.36(0.03)	0.04(0.01)	0.43(0.11)	You et al.
	YLD	0.12(0.05)	0.08(0.03)	0.24(0.05)	-0.05(0.02)	0.22(0.11)	Gaussian
EV	IOD	0.54(0.06)	0.42(0.02)	0.67(0.03)	-0.08(0.01)	0.70(0.11)	You et al.
	LIO	0.52(0.05)	0.62(0.04)	0.68(0.03)	-0.14(0.07)	0.70(0.11)	You et al.
	LIN	0.52(0.11)	0.65(0.03)	0.68(0.03)	-0.12(0.05)	0.70(0.11)	You et al.
	OIL	0.48(0.07)	0.33(0.03)	0.42(0.06)	-0.11(0.01)	0.56(0.15)	You et al.
	YLD	0.26(0.12)	0.25(0.06)	0.25(0.07)	0.15(0.02)	0.25(0.19)	Linear Basis
SU	IOD	0.46(0.09)	-0.22(0.05)	0.36(0.07)	-0.032(0.02)	0.40(0.19)	Linear Basis
	LIO	0.47(0.07)	0.34(0.07)	0.46(0.06)	0.33(0.03)	0.46(0.18)	Linear Basis
	LIN	0.45(0.06)	0.05(0.05)	0.42(0.07)	0.18(0.04)	0.43(0.19)	Linear Basis
	OIL	0.06(0.1)	0.19(0.03)	0.27(0.07)	0.06(0.01)	0.31(0.21)	You et al.
	YLD	0.19(0.1)	0.13(0.05)	0.21(0.05)	0.17(0.01)	0.29(0.23)	You et al.

have positive associations in both EV and SU populations, i.e., higher heritability corresponds to better accuracy. This is not the same in the BM population: For example, Gaussian RVM gives better result in OIL or YLD than in IOD, which has higher heritability than both.

7.3.1 Comparison with Related Work

Previously, You et al. [104] used three linear methods, Ridge Regression Best Linear Unbiased Prediction (RR-BLUP), Bayesian LASSO, and Bayesian ridge regression, over the three populations. They used broad-sense heritability of traits estimated in the three populations for building their models. They ran 5-fold cross-validation 500 times to evaluate the prediction accuracy of the five traits, IOD, LIO,

LIN, OIL, and YLD. Their obtained accuracies are shown in Table 7.1. However, they did not express which of the three linear models is the source of their obtained accuracies. It is not clear if all relate to one model, or each come from a different model (maximum of three accuracies). Nevertheless, comparing our results and You et al.'s shows that the RVM models are highly stable, as the standard deviations of RVM results are small compared to the standard deviation of the models chosen by You et al.. Also, there is significant superiority of Gaussian RVM in (BM, LIO) compared with the linear models (You et al.'s models and linear RVMs). Besides, N-gram RVM has comparable accuracies to the linear models in LIO trait in both BM and SU populations, though not the best. Definitely, such results imply non-linear relationships in some traits such as LIO, which cannot be captured by linear models. Thus for analysing this dataset, employing or constructing other non-linear kernels would be beneficial. Obtaining better results in such cases would not be unexpected, particularly for the LIO trait in the BM and SU populations.

7.4 Identifying Influential Markers

For identifying the most influential markers (SSR) on the five traits, we used our ensemble RVM architecture for ranking markers over the three populations. Each of these ensembles is composed of 100 linear basis RVMs, each with subsampling 80% of training data. The ensemble RVMs in each of the five traits ranked almost 100% of the BM markers, 87% of the EV markers, and 90% of the SU markers with

Table 7.2: Most influential markers recognized by ensemble RVMs

Population	Trait	Most significant markers
BM	IOD	Lu91, Lu223, Lu2909, Lu2163, Lu2555, Lu869, Lu502, Lu60, Lu747b, Lu2161, Lu3216
	LIO	Lu265, Lu652, Lu1146a, Lu461, Lu2032, Lu2974, Lu220, Lu91, Lu2825b, Lu1039, Lu2010b
	LIN	Lu223, Lu2909, Lu209, Lu2288, Lu91, Lu2163, Lu3238, Lu869, Lu3201, Lu60, Lu502
	OIL	Lu1049, Lu2340, Lu60, Lu2909, Lu3082, Lu3068, Lu2968, Lu2120a, Lu3113, Lu638, Lu3057a
	YLD	Lu91, Lu223, Lu869, Lu2909, Lu502, Lu2163, Lu2555, Lu2288, Lu209, Lu747b, Lu3201
EV	IOD	Lu2571, Lu803, Lu870, Lu2758, Lu3038, Lu2597, Lu2832, Lu943, Lu2513, Lu462a, Lu3016
	LIO	Lu2571, Lu803, Lu1169, Lu2010a, Lu2031, Lu3016, Lu2065, Lu2810, Lu2832, Lu870
	LIN	Lu2571, Lu803, Lu870, Lu2513, Lu2832, Lu1169, Lu2031, Lu943, Lu2810, Lu2827, Lu3016, Lu2010a
	OIL	Lu2120a, Lu2493, Lu52, Lu2863, Lu3085, Lu3017, Lu465, Lu989, Lu2286, Lu3157, Lu2449
	YLD	Lu2519, Lu2493, Lu2265, Lu2513, Lu628, Lu2865, Lu1094, Lu2923, Lu2582, Lu2430, Lu2219
SU	IOD	fad3B_207, Lu359_0, Lu2003_281, fad3A_84, Lu58a_257, Lu1052_0, Lu3218_497, Lu213_61, Lu2794_160, Lu2732_137
	LIO	fad3B_207, fad3A_84, Lu359_0, Lu58a_257, Lu2794_160, Lu2003_281, Lu213_61, Lu1052_0, Lu44E4_84, Lu2732_137
	LIN	fad3B_207, fad3A_84, Lu359_0, Lu58a_257, Lu2003_281, Lu2287_35, Lu213_61, Lu2794_160, Lu2732_137, Lu1052_0
	OIL	Lu1001_74, Lu3026_287, Lu998_275, Lu3281_865, Lu2262_31, Lu3266_733, Lu3097_338, Lu2794_160, Lu3068_313, Lu906_1
	YLD	Lu1077_174, Lu2162_7, Lu52_0, Lu41_202, Lu2247_28, Lu3099_353, Lu628_78, Lu2950_232, Lu2206, Lu805_11, Lu910_4

rank values in the range [1, 100]. The top ten ranks in each pair of (population, trait) are shown in Table 7.2. These markers are in the top 2-3% of all markers. Also, we presented a visualization of the top ranked markers in three populations in Figures 7.1, 7.2, and 7.3. The charts are only to visualize commonality, since they are not based on positions in chromosomes. These visualizations demonstrate markers shared among different traits. Influential markers on a trait can indicate positive or negative effects on a trait (or, additive versus non-additive effects).

7.4.1 Comparison with Related Works

There are previous studies which provide genetic maps and QTL detection in flax [19, 83, 84].

Cloutier et al. [19] obtained a flax dataset similar to the SU population (i.e., by crossing of the same two lines in the SU populations). Using QTL analysis, they detected two major QTLs each for LIO, LIN, and IOD. The result of comparing

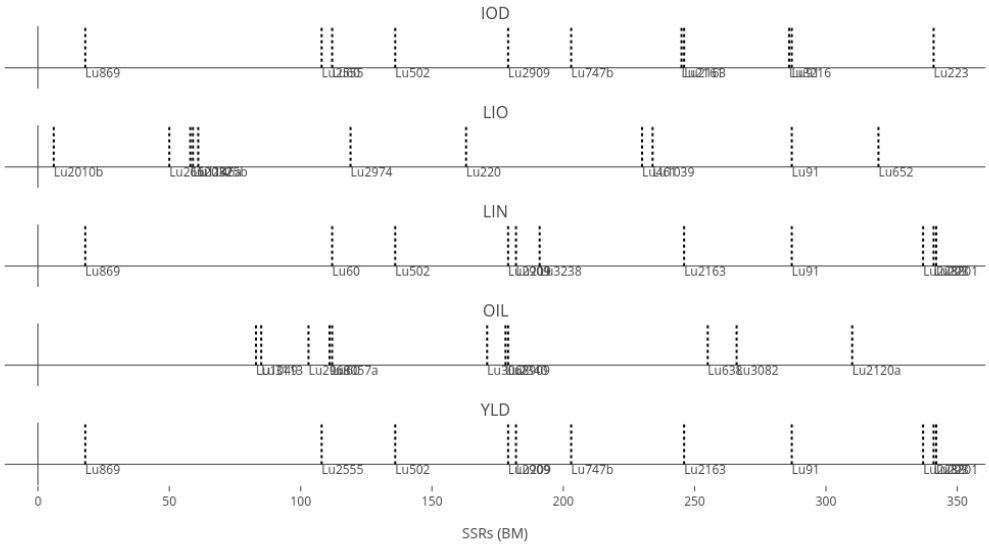


Figure 7.1: Most influential markers on the five traits in the BM population (Interactive Chart).

these QTLs with the top ranked markers identified by ensemble RVMs is shown in Table 7.3. [1] Marker $fad3A$, which is in the SU markers set, had been assigned top ranked for each of LIO, LIN, and IOD traits by the ensemble RVMs. This indicates that our results are valid. The second QTL they found was an interval between two markers which did not exist in our SU markers set. Therefore, we were not able to check the rank of the markers related to this QTL.

Soto-Cerda et al. [84] worked on a dataset, called the Canadian flax core collection, to find the markers associated with yield. Using association mapping, they

[1] To have a better view of ranks in Tables 7.4-7.3, consider that rank 34 and below in the BM population are in the top 10% of all markers. So are rank 44 and rank 47 in EV and SU populations, respectively.

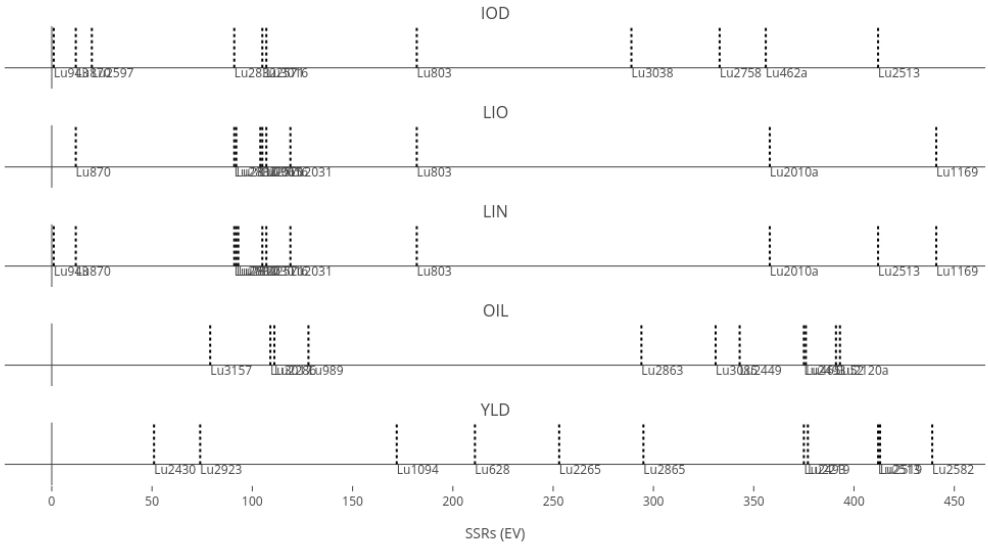


Figure 7.2: Most influential markers on the five traits in the EV population (Interactive Chart).

Table 7.3: QTLs detected by Cloutier et al. [19] versus influential markers recognized by RVM.

Marker	Trait	BM	EV	SU
fad3A	IOD, LIO, LIN	na	na	**IOD:rank4**, **LIO:rank2**, **LIN:rank2**, OIL:rank282, YLD:rank98
Lu206-Lu765B	IOD, LIO, LIN	na	na	na

identified 12 marker-trait associations for six agronomic traits, which they called yield-related traits: one thousand seed weight, seeds per boll, start of flowering, end of flowering, plant height, plant branching, and lodging. As Soto-Cerda et al. stated, these six traits may either directly affect yield, such as one thousand seed weight, or indirectly through adapting to regional growing conditions, thus avoiding yield and quality losses, such as flowering time. Table 7.4 shows a comparison of each of

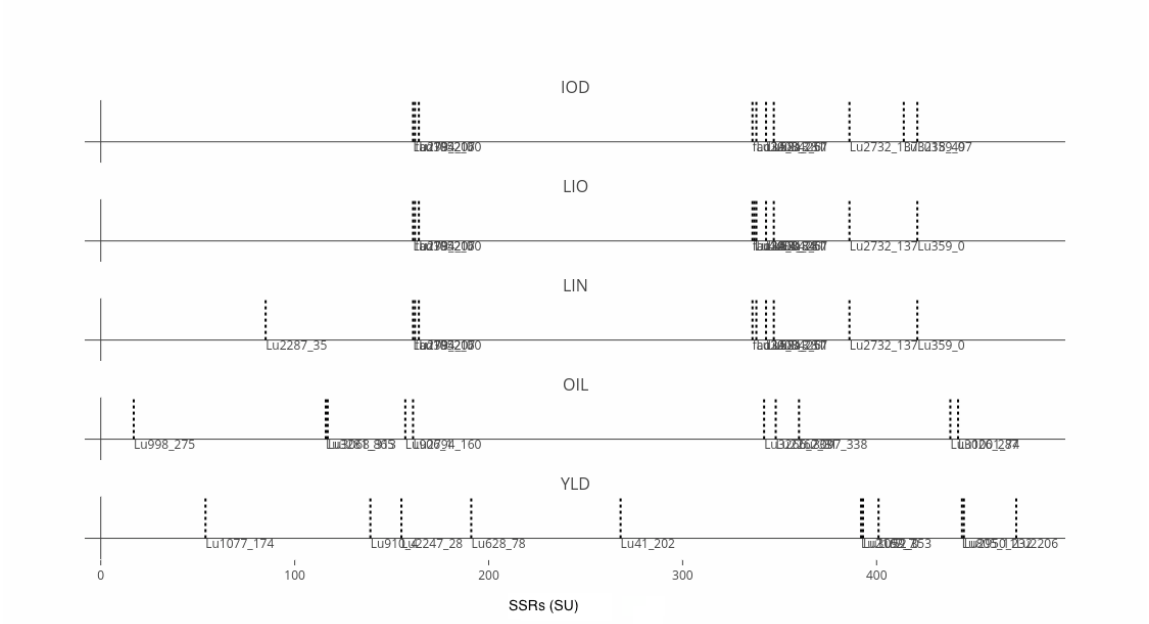


Figure 7.3: Most influential markers on the five traits in the SU population (Interactive Chart).

the markers identified by Soto-Cerda et al. to the rank of that marker (if it was in the population) assigned by our ensemble RVMs. For example, the marker Lu2555 associated with one thousand seed weight trait in Soto-Cerda et al.'s study, has high rank for traits IOD, LIN, and YLD in the BM population in our results (shown in boldface). In this example, the high rank of this marker in YLD matches their finding, but how do we interpret the high rank of the marker in IOD or LIN trait? It is possible that the marker Lu2555 is associated with a more general set of genes that are promoting general plant health. In particular, since the markers are not sufficiently dense it may be possible that the markers do not represent an individual gene but a collection of advantageous aspects of the genome.

We do not provide interpretations for all results, and we leave it to a biologist to get insight from them. For instance, suppose there is a relationship between the level of linoleic acid and seed weight. Hence, if a marker is significantly associated with the seed weight trait, then it will make sense if the same marker has association with linoleic acid. Also, this question might be raised that why does a marker has very different ranking in different populations (e.g., Lu2555 has high rank for YLD in the BM, but low rank in the EV). One reason for this happening can be difference in marker sets of populations. One marker might get lower relevance to a trait when it is along with some other markers. Also, a marker may just be representative for a set of advantageous aspects (e.g., SNPs) in the genome, and we have identified when these occur, but not what particular markers are affecting which genes. More dense data (such as SNPs) may be necessary to identify this.

Soto-Cerda et al. [83] used molecular diversity and association mapping to identify marker loci significantly associated with thousand seed weight, seeds per boll, dehiscent capsules, plant height, start of flowering, flower shape, petal color and petal overlap in pale and cultivated flax, respectively. We have listed the identified markers plus their ranks in our ensemble RVMs in Table 7.5 and 7.6.

7.5 Conclusion

In this chapter, we demonstrated how RVM performs in predicting seed yield, oil content, iodine value, linoleic, and linolenic acid content in flax. Until now, no state-

Table 7.4: Marker loci significantly associated with thousand seed weight (TSW), start of flowering (FL5), end of flowering (FL95), plant height (PH), plant branching (PB) and lodging (LDG) identified by Soto-Cerda et al. [84] versus influential markers on the five traits recognized by RVM.

Marker	Trait	BM	EV	SU
Lu2164	TSW	IOD:rank287, LIO:rank264, LIN:rank150, OIL:rank179, YLD:rank181	IOD:rank342, LIO:rank385, LIN:rank343, OIL:rank354, YLD:rank362	IOD:rank403, LIO:rank352, LIN:rank335, OIL:rank331, YLD:rank396
Lu2555	TSW	**IOD:rank5**, LIO:rank34, **LIN:rank12**, OIL:rank252, **YLD:rank7**	IOD:rank225, LIO:rank213, LIN:rank233, OIL:rank222, YLD:rank288	na
Lu2532	TSW	na	IOD:rank60, LIO:rank238, LIN:rank211, OIL:rank92, YLD:rank97	IOD:rank188, LIO:rank277, LIN:rank146, OIL:rank32, YLD:rank79
Lu58a	TSW	na	na	**IOD:rank5**, **LIO:rank4**, **LIN:rank4**, OIL:rank68, YLD:rank131
Lu526	TSW	IOD:rank29, LIO:rank202, LIN:rank23, OIL:rank294, YLD:rank35	na	na
Lu943	FL5, FL95,PH	IOD:rank317, LIO:rank179, LIN:rank153, OIL:rank75, YLD:rank196	**IOD:rank8**, LIO:rank17, **LIN:rank8**, OIL:rank43, YLD:rank59	na
Lu316	PH	na	na	na
Lu2067a	PB	IOD:rank146, LIO:rank283, LIN:rank300, OIL:rank334, YLD:rank108	IOD:rank377, LIO:rank348, LIN:rank334, OIL:rank217, YLD:rank242	IOD:rank396, LIO:rank432, LIN:rank357, OIL:rank300, YLD:rank438
Lu2560	LDG	IOD:rank282, LIO:rank246, LIN:rank169, OIL:rank321, YLD:rank229	na	na
Lu2564	LDG	IOD:rank241, LIO:rank41, LIN:rank228, OIL:rank295, YLD:rank216	IOD:rank107, LIO:rank228, LIN:rank36, OIL:rank205, YLD:rank103	na

of-the-art machine learning method such as RVM had been used for genomic selection in flax. We used a flax dataset [104] consisting of three populations with a full and a common set of SSR markers and some missing genotype data. We compared our phenotype prediction accuracies with existing classic linear methods accuracies [104] (such as BLUP) on the same dataset. We not only saw the superiority of RVM in some linear relationships, but also in some non-linear cases such as linoleic acid content and yield. We also investigated n-gram kernels in RVM, and saw their poor performances, similar to the yeast dataset, likely due to genetic linkage. However, we also witnessed the positive potential of n-gram kernels in identifying non-linear relationships in linoleic acid predictions.

We also investigated the relationship between broad-sense heritability and RVM

Table 7.5: Marker loci significantly associated with thousand seed weight (TSW), seeds per boll (SPB), dehiscent capsules (DEH), plant height (PH), start of flowering (FL5), flower shape (FS), petal color (PC) and petal overlap (PO) in pale flax identified by Soto-Cerda et al. [83] versus influential markers on the five traits recognized by RVM.

Marker	Trait	BM	EV	SU
Lu451b	TSW	na	na	na
Lu652	TSW	IOD:rank57, **LIO:rank2**, LIN:rank146, OIL:rank107, YLD:rank68	na	IOD:rank209, LIO:rank265, LIN:rank192, OIL:rank369, YLD:rank175
Lu1171	SPB	IOD:rank54, LIO:rank26, LIN:rank174, OIL:rank54, YLD:rank45	IOD:rank402, LIO:rank389, LIN:rank400, OIL:rank435, YLD:rank401	IOD:rank314, LIO:rank218, LIN:rank230, OIL:rank346, YLD:rank184
Lu2344	DEH	IOD:rank97, LIO:rank13, LIN:rank14, OIL:rank151, YLD:rank80	IOD:rank122, LIO:rank180, LIN:rank138, OIL:rank60, YLD:rank276	IOD:rank99, LIO:rank201, LIN:rank237, OIL:rank73, YLD:rank86
Lu442a	DEH	na	IOD:rank171, LIO:rank83, LIN:rank227, OIL:rank267, YLD:rank251	na
Lu265	PH	IOD:rank19, **LIO:rank1**, LIN:rank135, OIL:rank167, YLD:rank16	IOD:rank49, LIO:rank118, LIN:rank162, OIL:rank120, YLD:rank88	na
Lu271	FL5	na	IOD:rank415, LIO:rank411, LIN:rank424, OIL:rank428, YLD:rank387	IOD:rank386, LIO:rank370, LIN:rank255, OIL:rank104, YLD:rank18
Lu2344	FS	IOD:rank97, LIO:rank13, LIN:rank14, OIL:rank151, YLD:rank80	IOD:rank122, LIO:rank180, LIN:rank138, OIL:rank60, YLD:rank276	IOD:rank99, LIO:rank201, LIN:rank237, OIL:rank73, YLD:rank86
Lu2725	PC	IOD:rank18, LIO:rank155, LIN:rank15, OIL:rank279, YLD:rank14	na	na

Table 7.6: Marker loci significantly associated with thousand seed weight (TSW), seeds per boll (SPB), dehiscent capsules (DEH), plant height (PH), start of flowering (FL5), flower shape (FS), petal color (PC) and petal overlap (PO) in cultivated flax identified by Soto-Cerda et al. [83] versus influential markers on the five traits recognized by RVM.

Marker	Trait	BM	EV	SU
Lu2042	TSW	IOD:rank34, LIO:rank43, LIN:rank64, OIL:rank46, YLD:rank26	IOD:rank45, LIO:rank29, LIN:rank33, OIL:rank287, YLD:rank70	IOD:rank16, LIO:rank31, LIN:rank20, OIL:rank120, YLD:rank409
Lu2067a	PH,FL5	IOD:rank146, LIO:rank283, LIN:rank300, OIL:rank334, YLD:rank108	IOD:rank377, LIO:rank348, LIN:rank334, OIL:rank217, YLD:rank242	IOD:rank396, LIO:rank432, LIN:rank357, OIL:rank300, YLD:rank438
Lu943	FL5	IOD:rank317, LIO:rank179, LIN:rank153, OIL:rank75, YLD:rank196	**IOD:rank8**, LIO:rank17, **LIN:rank8**, OIL:rank43, YLD:rank59	na
Lu3038	PO	IOD:rank260, LIO:rank233, LIN:rank282, OIL:rank325, YLD:rank303	IOD:rank4, LIO:rank30, LIN:rank16, OIL:rank201, YLD:rank125	na

prediction accuracies. The results indicated that there is positive association between trait heritability and accuracies in two of three populations, so we have trait predictability in these populations, similar to the yeast dataset. In other words, the predictors are more successful in predicting traits with higher heritability than traits

with low heritability, but why? Why is trait predicting less successful for traits with lower heritability? For example, yield is a trait with low-heritability. It is said that yield is the most complex trait in crops [84]. Correspondingly, it is hard to identify important yield-related markers. We believe this complexity mainly comes from the ambiguity hidden in the definition of the trait itself. The yield of a crop such as flax is measured in weight unit per land unit (kilogram per hectare). We can see correlations between yield and other high heritability traits such as seed weight, seeds per boll, bolls per area, and even fatty acid oil contents. Another example akin to yield to some extent is oil content with intermediate heritability. Apparently, there is correlation between oil content and two high heritability traits, linoleic and linolenic acid. We think one approach to get better accuracies for lower heritability traits, such as yield, can be constructing models which integrate higher heritability trait in addition to the trait itself and variant sequences for prediction in future research.

In the last part of this chapter, we demonstrated how our RVM ensembles rank the markers in each of five traits in the three populations. We compared our identified top ranked markers to the limited existing markers influential on fatty acid composition traits and some agronomic traits and recognized by classical QTL analysis and association mapping. Interestingly, we found matches which validate our results, including makers *fad3A* [19] (influential on iodine, linoleic, and linolenic acid) and Lu2555 [84] (influential on seed weight which is correlated with yield). This indicates that our newly identified markers present a good set of candidates for further

investigation by a biologist.

Chapter 8

Conclusion

In this thesis, we first addressed the difficulties of two problems, genomic selection and identification of genome-wide associations of a complex trait. Then, having one simulated and two real-world datasets (yeast and flax), we showed how sparse Bayesian learning or RVM as a kernel-based method, with its unique advantages compared to SVM, can help us in these two problems.

For the first time, we employed RVMs with linear/non-linear kernels for predicting phenotypes via regression or classification. We showed that using ensemble RVM and bagging technique allow us to rank the RVs, and at the same time, improve the prediction accuracy and/or handle imbalanced data. We showed that if only prediction accuracy is our concern, then RVM can be considered as good as other successful learning methods. We also provided analyses of the RVs ranked by a trained ensemble RVM, and showed that how a set of top ranked RVs has the most important

individuals including both the "best" and "worst" genomes which a biologist might get insight from. We demonstrated that the identified RVs are not only much sparser than their counterparts in SVM (i.e., SVs), but also better representatives of data.

A major contribution of our work is to define sparse Bayesian learning in such a way that we can discriminate between kernel and basis functions, i.e., "kernel" RVM versus "basis" RVM. We introduced a new approach based on linear basis RVM, ensemble method, and bagging technique for feature selection and ranking. We showed that how this framework can help us to find most influential markers of a complex trait, as well as non-relevant markers. We also presented the sensitivity analysis of model parameters, i.e., bootstrap size and number of base learner in an ensemble versus prediction accuracy.

Our ensemble linear basis RVM is an embedded method in which feature selection is part of the model building. We compared our results with the existing results from the traditional QTL mapping and association analysis in our real datasets, and demonstrated the validity of our approach. However, our method also found some new influential makers in both flax and yeast that can be considered by a biologist for further investigations. As we showed in the simulated dataset, our proposed method is capable of recognizing markers with any sort of effect, either additive or non-additive. We also discussed about heritability and found that we have higher trait predictability in traits with high heritability than traits with low heritability.

Future Work

We think the next steps can be in three directions: (1) Gaussian basis RVMs for feature selection, (2) Sequence kernels for genotypic data, and (3) Predictors for low-heritability traits.

Gaussian basis RVMs: Our ensemble linear basis RVM for feature selection takes in to account only linear relationships. Although this linear separability is a reasonable assumption for high dimensional data, it is desirable to try nonlinear basis substitution, particularly Gaussian function, to handle nonlinear relationships. Gaussian basis RVM still gives feature RVs as each Gaussian basis in the model operates on a different dimension (feature). However, employing Gaussian basis RVM requires setting not only the variance (σ_m) in each Gaussian basis function in (2.5), but also the mean or center (μ_m):

$$\phi_m(\mathbf{x}) = \exp(-\frac{(\mathbf{x}^{[m]} - \mu_m)^2}{{\sigma_m}^2}),$$

where $\mathbf{x}^{[m]}$ refers to the m-th feature in an input vector $\mathbf{x}$ with M dimensions. Investigating any appropriate approach, with acceptable computational complexity, for choosing parameters in Gaussian basis RVMs, and employing these RVMs in an application remain as future work.

Sequence kernels for genotypic data: Our other contribution was training RVMs with sequence rational kernels including n-gram kernels, that had never investigated in any other applications. We observed that the sequence kernel RVMs presented poor predictions on both yeast and flax datasets. We provided an analysis indicating that the poor performance is due to genetic linkage, and not because of RVM. Therefore, we suggest sequence kernels which are not affected by genetic linkage be investigated in future for genomic selection.

Predictors for low heritability traits: We also discussed about heritability and found that we have lower trait predictability in traits with lower heritability such as yield. This result was not specific to RVM predictors, but corresponds to other computational methods that previously had been used. As any low heritability trait (e.g., yield) has correlations with one or more high heritability traits (e.g., yield with seed weight), we propose constructing predictive models which also capture the dependencies of the trait to the traits with higher heritability, in order to improve accuracies. Finding such solutions remain as future work.

Bibliography

[1] G. Abraham and M. Inouye. Genomic risk prediction of complex human disease and its clinical application. *Current opinion in genetics & development*, 33:10–16, 2015.

[2] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. Openfst: A general and efficient weighted finite-state transducer library. In *International Conference on Implementation and Application of Automata*, pages 11–23. Springer, 2007.

[3] C. Allauzen, M. Mohri, and A. Talwalkar. Sequence kernels for predicting protein essentiality. In *Proceedings of the 25th international conference on Machine learning*, pages 9–16. ACM, 2008.

[4] C. Allauzen, M. Mohri, and A. Rostamizadeh. OpenKernel Library. `http://www.openkernel.org`, 2018.

[5] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri. OpenFST Library. `http://www.openfst.org/twiki/bin/view/FST/WebHome`, 2018.

[6] J.-Y. An, F.-R. Meng, Z.-H. You, Y.-H. Fang, Y.-J. Zhao, and M. Zhang. Using the relevance vector machine model combined with local phase quantization to predict protein-protein interactions from protein sequences. *BioMed Research International*, 2016, 2016.

[7] A. Anaissi, M. Goyal, D. R. Catchpoole, A. Braytee, and P. J. Kennedy. Ensemble feature learning of genomic data using support vector machine. *PloS one*, 11(6):e0157330, 2016.

[8] M. L. Avolio, J. M. Beaulieu, E. Y. Lo, and M. D. Smith. Measuring genetic diversity in ecological studies. *Plant Ecology*, 213(7):1105–1115, 2012.

[9] P. Baldi and S. Brunak. *Bioinformatics: The machine learning approach*. MIT Press, Cambridge, MA, USA, 2nd edition, 2001.

[10] R. Batuwita and V. Palade. Class Imbalance Learning Methods for Support Vector Machines. In *Imbalanced Learning: Foundations, Algorithms, and Applications*, pages 83–99. John Wiley & Sons, 2013.

[11] A. Ben-Hur, C. S. Ong, S. Sonnenburg, B. Schölkopf, and G. Rätsch. Support vector machines and kernels for computational biology. *PLoS Comput Biol*, 4 (10):e1000173, 2008.

[12] C. M. Bishop and M. E. Tipping. Variational relevance vector machines. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pages 46–53. Morgan Kaufmann Publishers Inc., 2000.

[13] M. Blondel, A. Onogi, H. Iwata, and N. Ueda. A ranking approach to genomic selection. *PloS one*, 10(6):e0128570, 2015.

[14] J. S. Bloom, I. M. Ehrenreich, W. T. Loo, T.-L. V. Lite, and L. Kruglyak. Finding the sources of endfor heritability in a yeast cross. *Nature*, 494(7436): 234, 2013.

[15] L. Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[16] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[17] W. S. Bush and J. H. Moore. Genome-wide association studies. *PLoS computational biology*, 8(12):e1002822, 2012.

[18] G. C. Cawley and N. L. Talbot. Gene selection in cancer classification using sparse logistic regression with bayesian regularization. *Bioinformatics*, 22(19): 2348–2355, 2006.

[19] S. Cloutier, R. Ragupathy, Z. Niu, and S. Duguid. SSR-based linkage map of flax (linum usitatissimum l.) and mapping of qtls underlying fatty acid composition traits. *Molecular Breeding*, 28(4):437–451, 2011.

[20] P. G. C. C. Committee. Genomewide association studies: history, rationale, and prospects for psychiatric disorders. *Focus*, 8(3):417–434, 2010.

[21] C. Cortes and M. Mohri. Learning with weighted transducers. In *Proceedings of the 2009 conference on Finite-State Methods and Natural Language Processing:*

Post-proceedings of the 7th International Workshop FSMNLP 2008, pages 14–22. IOS Press, 2009.

[22] C. Cortes, P. Haffner, and M. Mohri. Rational kernels: Theory and algorithms. *Journal of Machine Learning Research*, 5(Aug):1035–1062, 2004.

[23] C. Cortes, M. Mohri, and A. Rostamizadeh. Learning sequence kernels. In *2008 IEEE Workshop on Machine Learning for Signal Processing*, pages 2–8. IEEE, 2008.

[24] Z. A. Desta and R. Ortiz. Genomic selection: genome-wide prediction in plant improvement. *Trends in plant science*, 19(9):592–601, 2014.

[25] T. G. Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[26] T. A. Down and T. J. Hubbard. Relevance vector machines for classifying points and regions in biological sequences. *arXiv preprint q-bio/0312006*, 2003.

[27] B. Efron and R. J. Tibshirani. *An introduction to the bootstrap*. CRC press, 1994.

[28] E. Eskin and S. Snir. The homology kernel: a biologically motivated sequence embedding into euclidean space. In *2005 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, pages 1–8. IEEE, 2005.

[29] Flax Council of Canada. https://flaxcouncil.ca/resources/about-flax/canada-a-flax-leader/, 2018.

[30] Flax Council of Canada. https://flaxcouncil.ca/wp-content/uploads/2015/03/FlxPrmr_4ed_Chpt1.pdf, 2018.

[31] Flax Council of Canada. https://flaxcouncil.ca/flax-usage/industrial-uses/fibre-uses/, 2018.

[32] S. K. Forsberg, J. S. Bloom, M. J. Sadhu, L. Kruglyak, and Ö. Carlborg. Accounting for genetic interactions improves modeling of individual quantitative trait phenotypes in yeast. *Nature genetics*, 49(4):497, 2017.

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

[34] A. García-Ruiz, J. B. Cole, P. M. VanRaden, G. R. Wiggans, F. J. Ruiz-López, and C. P. Van Tassell. Changes in genetic selection differentials and generation intervals in us holstein dairy cattle as a result of genomic selection. *Proceedings of the National Academy of Sciences*, 113(28):E3995–E4004, 2016.

[35] P. Giraldo, C. Royo, M. González, J. M. Carrillo, and M. Ruiz. Genetic diversity and association mapping for agromorphological and grain quality traits of a structured collection of durum wheat landraces including subsp. durum, turgidum and diccocon. *PloS one*, 11(11):e0166577, 2016.

[36] J. M. González-Camacho, L. Ornella, P. Pérez-Rodríguez, D. Gianola, S. Dreisigacker, and J. Crossa. Applications of machine learning methods to genomic selection in breeding wheat for rust resistance. *The Plant Genome*, 2018.

[37] O. González-Recio, G. J. Rosa, and D. Gianola. Machine learning methods and predictive ability metrics for genome-wide prediction of complex traits. *Livestock Science*, 166:217–231, 2014.

[38] N. F. Grinberg and R. D. King. An evaluation of machine-learning for predicting phenotype: studies in yeast and wheat. *bioRxiv*, page 105528, 2017.

[39] N. F. Grinberg, A. Lovatt, M. Hegarty, A. Lovatt, K. P. Skøt, R. Kelly, T. Blackmore, D. Thorogood, R. D. King, I. Armstead, et al. Implementation of genomic prediction in lolium perenne (l.) breeding populations. *Frontiers in plant science*, 7, 2016.

[40] N. F. Grinberg, O. I. Orhobor, and R. D. King. An evaluation of machine-learning for predicting phenotype: Studies in yeast, rice and wheat. *bioRxiv*, page 105528, 2018.

[41] Y. Guo, Z. Wei, B. J. Keating, and H. Hakonarson. Machine learning derived risk prediction of Anorexia Nervosa. *BMC medical genomics*, 9(1):4, 2016.

[42] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

[43] D. Habier, R. L. Fernando, and D. J. Garrick. Genomic-blup decoded: a look into the black box of genomic prediction. *Genetics*, pages genetics–113, 2013.

[44] Z. M. Hira and D. F. Gillies. A review of feature selection and feature extraction methods applied on microarray data. *Advances in bioinformatics*, 2015, 2015.

[45] S. Hoban. An overview of the utility of population simulation software in molecular ecology. *Molecular ecology*, 23(10):2383–2401, 2014.

[46] T. Hofmann, B. Schölkopf, and A. J. Smola. Kernel methods in machine learning. *The annals of statistics*, pages 1171–1220, 2008.

[47] N. H. G. R. Institute. Genome-wide association studies. `https://www.genome.gov/20019523/genomewide-association-studies-fact-sheet/`.

[48] J.-L. Jannink, A. J. Lorenz, and H. Iwata. Genomic selection in plant breeding: from theory to practice. *Briefings in functional genomics*, 9(2):166–177, 2010.

[49] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

[50] K. E. Kemper and M. E. Goddard. Understanding and predicting complex traits: knowledge from cattle. *Human molecular genetics*, 21(R1):R45–R51, 2012.

[51] J. W. Kimball. Online Biology Book: Imprinted Genes. `http://www.biology-pages.info/I/Imprinting.html`.

[52] R. I. Kondor and J. Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *Proceedings of the 19th international conference on Machine learning*, pages 315—122, 2002.

[53] A. Korte and A. Farlow. The advantages and limitations of trait analysis with GWAS: a review. *Plant methods*, 9(1):29, 2013.

[54] J. Kruppa, A. Ziegler, and I. R. König. Risk estimation and risk prediction using machine-learning methods. *Human genetics*, 131(10):1639–1654, 2012.

[55] C. S. Leslie, E. Eskin, A. Cohen, J. Weston, and W. S. Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467–476, 2004.

[56] L. Li, Y. Long, L. Zhang, J. Dalton-Morgan, J. Batley, L. Yu, J. Meng, and M. Li. Genome wide analysis of flowering time trait in multiple environments via high-throughput genotyping technique in Brassica napus L. *PloS one*, 10 (3):e0119425, 2015.

[57] Y. Li, C. Campbell, and M. Tipping. Bayesian automatic relevance determination algorithms for classifying gene expression data. *Bioinformatics*, 18(10): 1332–1339, 2002.

[58] Y. Li, K. K. Lee, S. Walsh, C. Smith, S. Hadingham, K. Sorefan, G. Cawley, and M. W. Bevan. Establishing glucose- and ABA-regulated transcription

networks in Arabidopsis by microarray analysis and promoter classification using a Relevance Vector Machine. *Genome research*, 16(3):414–427, 2006.

[59] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2 (Feb):419–444, 2002.

[60] T. A. Manolio, F. S. Collins, N. J. Cox, D. B. Goldstein, L. A. Hindorff, D. J. Hunter, M. I. McCarthy, E. M. Ramos, L. R. Cardon, A. Chakravarti, et al. Finding the missing heritability of complex diseases. *Nature*, 461(7265):747, 2009.

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

[62] I. Miko and L. LeJeune, editors. *Essentials of Genetics*, chapter 3.4. Cambridge, MA: NPG Education, 2009.

[63] C. Miles and M. Wayne. Quantitative trait locus (qtl). *Nature education*, 1(1): 208, 2008.

[64] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT Press, 2012.

[65] G. Moser, B. Tier, R. E. Crump, M. S. Khatkar, and H. W. Raadsma. A

comparison of five methods to predict genomic breeding values of dairy bulls from genome-wide snp markers. *Genetics Selection Evolution*, 41(1):56, 2009.

[66] S. Mucha, M. Pszczoła, T. Strabel, A. Wolc, P. Paczyńska, and M. Szydlowski. Comparison of analyses of the qtlmas xiv common dataset. ii: Qtl analysis. In *BMC proceedings*, volume 5, page S2. BioMed Central, 2011.

[67] C. J. Needham, J. R. Bradford, A. J. Bulpitt, and D. R. Westhead. A primer on learning in bayesian networks for computational biology. *PLoS computational biology*, 3(8):e129, 2007.

[68] S. W. G. of the Psychiatric Genomics Consortium et al. Biological insights from 108 schizophrenia-associated genetic loci. *Nature*, 511(7510):421–427, 2014.

[69] J. O. Ogutu, H.-P. Piepho, and T. Schulz-Streeck. A comparison of random forests, boosting and support vector machines for genomic selection. In *BMC proceedings*, volume 5, page 1. BioMed Central, 2011.

[70] S. Okser, T. Pahikkala, A. Airola, T. Salakoski, S. Ripatti, and T. Aittokallio. Regularized machine learning in the genetic prediction of complex traits. *PLoS Genet*, 10(11):e1004754, 2014.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn:

Machine Learning in Python . *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

[72] J. Poland and J. Rutkoski. Advances and challenges in genomic selection for disease resistance. *Annual review of phytopathology*, 54:79–98, 2016.

[73] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM journal of research and development*, 3(2):114–125, 1959.

[74] A. Roche-Lima, M. Domaratzki, and B. Fristensky. Metabolic network prediction through pairwise rational kernels. *BMC bioinformatics*, 15(1):1, 2014.

[75] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[76] Y. Saeys, T. Abeel, and Y. Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 313–325. Springer, 2008.

[77] C. Sandor and M. Georges. On the detection of imprinted qtl in line crosses: Effect of linkage disequilibrium. *Genetics*, 2008.

[78] R. E. Schapire and Y. Freund. *Boosting: Foundations and algorithms*. 2012.

[79] B. Schölkopf, I. Guyon, J. Weston, P. Frasconi, and R. Shamir. Statistical

learning and kernel methods in bioinformatics. *Nato Science Series Sub Series III Computer and Systems Sciences*, (183):1–21, 2003.

[80] X. Shen. The curse of the missing heritability. *Frontiers in Genetics*, 4:225, 2013.

[81] A. Shmilovici and D. Ben-Shimon. A feature selection strategy for the relevance vector machine. *Recent Advances in Knowledge Engineering and Systems Science*, pages 73–78, 2013.

[82] S. Sonnenburg, G. Rätsch, and K. Rieck. Large scale learning with string kernels. chapter 4. MIT Press, 2007.

[83] B. J. Soto-Cerda, A. Diederichsen, S. Duguid, H. Booker, G. Rowland, and S. Cloutier. The potential of pale flax as a source of useful genetic variation for cultivated flax revealed through molecular diversity and association analyses. *Molecular breeding*, 34(4):2091–2107, 2014.

[84] B. J. Soto-Cerda, S. Duguid, H. Booker, G. Rowland, A. Diederichsen, and S. Cloutier. Genomic regions underlying agronomic traits in linseed (linum usitatissimum l.) as revealed by association mapping. *Journal of integrative plant biology*, 56(1):75–87, 2014.

[85] G. Stiglic, S. Kocbek, I. Pernek, and P. Kokol. Comprehensive decision tree models in bioinformatics. *PloS one*, 7(3):e33812, 2012.

[86] G. Su, O. F. Christensen, T. Ostersen, M. Henryon, and M. S. Lund. Estimating additive and non-additive genetic variances and predicting genetic merits using genome-wide dense single nucleotide polymorphism markers. *PloS one*, 7(9):e45293, 2012.

[87] M. Szydlowski, editor. *BMC Proceedings of the 14th European workshop on QTL mapping and marker assisted selection (QTL-MAS)*, volume 5, 2011.

[88] M. Szydlowski and P. Paczyńska. Qtlmas 2010: simulated dataset. In *BMC proceedings*, volume 5, page S3. BioMed Central, 2011.

[89] S. Szymczak, J. M. Biernacka, H. J. Cordell, O. González-Recio, I. R. König, H. Zhang, and Y. V. Sun. Machine learning in genome-wide association studies. *Genetic epidemiology*, 33(S1):S51–S57, 2009.

[90] M. E. Tipping. V2.0 SparseBayes Software for Matlab. `http://www.miketipping.com/sparsebayes.htm`.

[91] M. E. Tipping. The relevance vector machine. pages 652–658. MIT Press, 2000.

[92] M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001.

[93] M. E. Tipping, A. C. Faul, et al. Fast marginal likelihood maximisation for

sparse Bayesian models. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

[94] V. N. Vapnik. *Statistical learning theory*, volume 1. Wiley New York, 1998.

[95] L. Wang, Y. Wang, and Q. Chang. Feature selection methods for big data bioinformatics: A survey from the search perspective. *Methods*, 111:21–31, 2016.

[96] Z. Wei, K. Wang, H.-Q. Qu, H. Zhang, J. Bradfield, C. Kim, E. Frackleton, C. Hou, J. T. Glessner, R. Chiavacci, et al. From disease association to risk assessment: an optimistic view from genome-wide association studies on type 1 diabetes. *PLoS Genet*, 5(10):e1000678, 2009.

[97] E. W. Weisstein. Ill-conditioned matrix From MathWorld—A Wolfram Web Resource. `http://mathworld.wolfram.com/Ill-ConditionedMatrix.html`.

[98] N. Wray and P. Visscher. Estimating trait heritability. *Nature education*, 1(1): 29, 2008.

[99] C.-C. Wu, S. Asgharzadeh, T. J. Triche, and D. Z. D'Argenio. Prediction of human functional genetic networks from heterogeneous data using RVM-based ensemble learning. *Bioinformatics*, 26(6):807–813, 2010.

[100] Y. Xiao, H. Liu, L. Wu, M. Warburton, and J. Yan. Genome-wide association studies in maize: praise and stargaze. *Molecular plant*, 10(3):359–374, 2017.

[101] P. Yang, Y. H. Yang, B. B. Zhou, and A. Y. Zomaya. A review of ensemble methods in bioinformatics. *Current Bioinformatics*, 5(4):2383–2401, 2010.

[102] X. Yang, W. Pan, and Y. Guo. Sparse bayesian classification and feature selection for biological expression data with high correlations. *PloS one*, 12 (12):e0189541, 2017.

[103] C. Yao, X. Zhu, and K. A. Weigel. Semi-supervised learning for genomic prediction of novel traits with small reference populations: an application to residual feed intake in dairy cattle. *Genetics Selection Evolution*, 48(1):84, 2016.

[104] F. M. You, H. M. Booker, S. D. Duguid, G. Jia, and S. Cloutier. Accuracy of genomic selection in biparental populations of flax (*Linum usitatissimum L.*). *The Crop Journal*, 4(4):290–303, 2016.

[105] Y. Zhang and J. C. Rajapakse. *Machine learning in bioinformatics*, volume 4. John Wiley & Sons, 2009.

[106] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.