uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Robert Davies
AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.Sc. (Mathematics)
GRADE / DEGREE

Department of Mathematics
FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

On the Generation of a Classification Algorithm from DNA Based Microarray Studies
TITRE DE LA THÈSE / TITLE OF THESIS

G. Wells
DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

V. Pestov
CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

D. Bickel

S. Aris-Brosou

J. Nielsen

Gary W. Slater
Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

On the Generation of a Classification Algorithm from DNA Based Microarray Studies

Robert William Davies

Thesis Submitted to the Faculty of Graduate and Postdoctoral Studies
In partial fulfilment of the requirements for the degree of Master of Science in Mathematics [1]

Department of Mathematics and Statistics
Faculty of Science
University of Ottawa

[1]The M.Sc. program is a joint program with Carleton University, administered by the Ottawa-Carleton Institute of Mathematics and Statistics

Your file Votre référence
ISBN: 978-0-494-66230-4
Our file Notre référence
ISBN: 978-0-494-66230-4

Abstract

The purpose of this thesis is to build a classification algorithm using a Genome Wide Association (GWA) study. Briefly, a GWA is a case-control study using genotypes derived from DNA microarrays for thousands of people. These microarrays are able to acquire the genotypes of hundreds of thousands of Single Nucleotide Polymorphisms (SNPs) for a person at a time. In this thesis, we first describe the processes necessary to prepare the data for analysis. Next, we introduce the Naive Bayes classification algorithm and a modification so that effects of a SNP on the disease of interest are weighted by a Bayesian posterior probability of association. This thesis then uses the data from three coronary artery disease GWAs, one as a training set and two as test sets, to build and test the classifier. Finally, this thesis discusses the relevance of the results and the generalizability of this method to future studies.

Acknowledgements

I would like use this section to gratefully acknowledge the numerous people without whom this thesis would not be possible. First and foremost, I would like to thank my supervisor Dr. George Wells, for giving me great freedom in choosing and pursuing a thesis topic, for providing feedback and giving guidance. I would also like to thank my co-supervisor Dr. Vladimir Pestov, for helping to initiate my interest in machine learning and for many invigorating mathematical conversations.

I am also grateful for the study data which formed an integral part of my thesis. I would like to acknowledge the work of the members of the John and Jennifer Ruddy Canadian Cardiovascular Genetics Center from the University of Ottawa Heart Institute for the Ottawa Heart Genomics Study, particularly the leadership of Dr. Robert Roberts, Dr. Alexandre Stewart and Dr. Ruth McPherson, and the statistical work of Kathryn Williams and Li Chen. I am also grateful for access to the genome wide association studies provided by the Wellcome Trust Case Control Consortium and the Cleveland Clinic Foundation, and would like to thank all the people that made those studies possible. I would also like to thank Chantal Giroux and Lorraine Houle from the University of Ottawa Department of Mathematics and Statistics, and Anne Gray from the University of Ottawa Heart Institute, for making sure that I was always registered and properly supported while a student.

Finally, I would like to thank my friends at the University of Ottawa Heart Institute, who made my work environment very enjoyable.

Dedication

This thesis is dedicated to my parents. Mom, for consistently ensuring I was knowledgeable and aware of the rules and regulations of graduate studies; Dad, for getting me started in research at a young age.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis will focus on the application of techniques from machine learning to analyze the results from a large genetics study, the Ottawa Heart Genomics Study (OHGS), whose purpose is to investigate the genetic causes of Coronary Artery Disease (CAD). Specifically, it will focus on the issues related to the construction and implementation of a classification algorithm whose purpose is to assess likelihood of developing CAD based on genetic risk factors.

Epidemiologically, the OHGS is a Genome Wide Association (GWA) study featuring several thousand cases and controls, with cases having CAD and controls being asymptomatic for CAD. DNA is acquired from these individuals and analyzed using a DNA microarray, which returns a list of genotypes at each of several hundred thousand Single Nucleotide Polymorphisms (SNPs). These SNPs are common genetic variants, which are present in appreciable frequencies in a given population, usually at least 1%. If there is a difference between the frequencies of the SNPs in cases and controls, it is evidence that the genomic region surrounding the SNPs is associated with the disease phenotype, and may in fact be causal in its genesis.

In the OHGS, DNA microarrays are used which genotype half a million SNPs for a combined case and control study population of 2997 people. Traditional analysis of

GWA studies such as the OHGS involves analyzing the differences in SNP frequencies for each of the loci on the chip individually using single variable tests of association, and then correcting for multiple comparisons. This style of analysis, which is by far the most common method of analysis for GWA studies, is useful in detecting loci which influence disease; however, it is lacking in its ability to assess risk of disease. That is, while it can determine loci associated with disease, it does not address the issue of how these loci act to affect a person's risk of disease as a whole.

Therefore, the primary objective of this thesis is to build a classification algorithm for coronary artery disease using genetic data from a genome wide association study. This thesis will explore in detail the statistical framework behind GWAs and machine learning, as well as give results from an application of these methods to results from three real world studies.

This thesis will be organized as follows. Chapter 2, the Background chapter, will introduce all the relevant background from the literature which will be necessary to understand the remainder of the thesis. Note that a large portion of the Background chapter will focus on deriving or explaining concepts from first principles. This is not just exposition, but will serve one of the following purposes: either the derivation will be necessary to understand the later derivations of the Methods chapter; or the derivation will be necessary so that the concept could be coded in the Programs chapter; or that a knowledge of the derivation will be necessary to understand important realizations made in the Discussion chapter.

Chapter 3, the Methods chapter, will introduce two methods which were significantly expanded upon for the purposes of this thesis. They are the r^2 filter which is used to mitigate redundancy in the GWA dataset, and the Naive Bayes classifier, the classification algorithm which was selected for this thesis. Note that both the expansion of the derivation of the Naive Bayes classifier and the proof of its statistical consistency are new results which were derived for the purpose of this thesis.

Chapter 4, the Programs chapter, will serve as a bridge between the highly theo-

retical Background and Methods chapter, and the very applied Application chapter. This chapter will summarize the computer programs which were written to implement the methods of Chapters 2 and 3. These programs were mostly written *de novo* based on the contents of the Background and Methods chapters, partly to account for the new methods for which code does not exist, and partly to ensure the methods could work given the large size of the GWA dataset.

Chapter 5, the Application chapter, will introduce three GWA studies, the Ottawa Heart Genomics Study (OHGS), the Wellcome Trust Case Control Consortium (WTCCC); and the Cleveland Clinic Foundation (CCF). The chapter will then use the code from the Programs chapter to perform quality control and construct the Naive Bayes classifier from the datasets, while detailing the performance of the classifier.

Chapter 6, the Discussion chapter, will discuss both the results of the Application chapter and the various issues that arose during the undertaking of this thesis. This includes a discussion of other machine learning algorithms which were considered alongside the Naive Bayes algorithm, as well as discussing the results of the Application chapter and the potential clinical utility of methods such as presented in this thesis.

Chapter 7, the Conclusion chapter, will end the thesis with some concluding thoughts, and directions for future work.

Lastly, there will be two appendices. Appendix A will feature some of the notation and acronyms which are used commonly in this thesis. Appendix B will house some of the code from the most important components written in the Programs chapter.

Chapter 2

Background

The purpose of this Background chapter is to introduce the concepts from cardiology, genetics, mathematics, statistics and machine learning which will be necessary to understand this thesis. Whenever possible, material will be described from first principles; particularly the derivation of the relevant statistical material. As stated earlier in the Introduction, the purpose of this rigorous derivation is multi-fold, in that it was either necessary to understand the later derivations in the Methods chapter, necessary from a programming perspective for the Programs chapter, or it was necessary to understand some of the interpretations of the results in the Discussion chapter.

This chapter will be organized as follows. The first section will give a short primer on the biology of coronary artery disease and myocardial infarction. The second section will introduce the concepts from genetics which are necessary to understand this thesis, including DNA microarrays and genome wide association studies. The third section will discuss the statistical analysis of genome wide association studies. Finally, the fourth section will introduce machine learning and the topic of supervised learning.

Note that these background sections need not be read in order; the coronary

artery disease and machine learning sections are relatively stand-alone, however the best understanding should come from consecutive reading.

2.1 Coronary Artery Disease

The function of the heart is to circulate oxygenated and nutrient rich blood through the body; it also supplies itself, through the coronary arteries. Over time, the arteries of the body may accumulate plaque, a collection of macrophages (white blood cells), fat, cholesterol and various proteins and connective tissue, a process known as atherosclerosis. Coronary Artery Disease (CAD) occurs when these atherosclerotic plaques are deposited in the arteries feeding the heart.

CAD can lead to two major problems. The first is the gradual narrowing of the blood vessel to the point where blood flow becomes occluded, known as a stenosis. This can lead to an ischemic (lack of oxygen) condition in the heart tissue, where the muscle cannot contract normally. Subsequent to this, an individual may develop angina (chest pain), and suffer from a decreased cardiovascular output.

CAD is also problematic in that it is the most likely precursor to a Myocardial Infarction (MI), also known as a heart attack. Note that while CAD is almost always required for an MI, it is possible to have stable CAD which is not likely to trigger an MI. An MI itself can occur when one of the plaque deposits in a coronary artery becomes unstable, ruptures, and triggers red blood cell and platelet deposition as a blood clot. While blood clotting is a useful physiological response to injury, it can be devastating in the narrow confines of the coronary arteries. A clot in a coronary artery can severely or entirely block the vessel, cutting off blood supply to the downstream region of the heart. If not treated rapidly, this can lead to death of the affected tissue and severe impairment of heart function; in severe cases, it can cause death.

Given its potential to cause serious bodily harm, understanding CAD is an important matter of public health. While there exists a number of factors which have well-

established links to CAD and MI, such as diabetes, hypertension, hypercholesteremia (high cholesterol), sex, age and smoking, [7], the role that genetics plays in CAD risk is less clear. Understanding the impact of genetic factors in the pathogenesis of the disease will go a long way towards understanding, preventing and treating CAD.

2.2 Genetics

Genetics is the field of biology which concerns itself with hereditary traits. The purpose of this section will be to introduce the concepts from genetics which are essential to understanding the later mathematical analysis of genetic data.

This section will be organized as follows. The first subsection will introduce DNA and genetic variants known as Single Nucleotide Polymorphisms (SNPs). The second subsection will introduce DNA microarrays as methods for acquiring an individuals genotype at hundreds of thousands of SNPs. Lastly, the third subsection will discuss a particular type of epidemiological study, the genome wide association study, as a method for finding SNPs which are associated with a particular trait or disease.

2.2.1 DNA and SNPs

One of the fundamental tenets of cell theory, and the foundation for modern cell biology and genetics, is that all living organisms contain hereditary genetic material, which they obtain from their parent(s) and will pass to any offspring [3]. Molecularly, this hereditary genetic information is DeoxyriboNucleic Acid (DNA), which in humans is stored in a complementary double-stranded form. Before we consider the inheritance of DNA and its effects on human physiology, we need to study DNA the molecule.

DNA itself is a long polymer, composed of simple units known as nucleotides, which, for DNA, are Cytosine (C), Guanine (G), Adenosine (A) and Thymine (T).

For two strands of DNA to come together to form double stranded DNA, they must feature complementary bases at adjacent sites; Cytosine always pairs with Guanine, and Adenosine always pairs with Thymine. These complementary bindings are known as base pairs (bps), and form the basic unit of information on a strand of DNA. In humans, DNA is stored in cells in the form of chromosomes, which are single molecules of DNA, organized and coiled around certain proteins. The size of chromosomes varies in relation to their names, with chromosome 1 (250 million base pairs) being the largest and chromosome 22 (50 million base pairs) being the smallest. The sum of the genetic material across the chromosomes, including both the sex and the autosomal chromosomes, is approximately 3 billion base pairs in humans.

During the transmission of information from one generation to the next, errors in DNA copying or transmission may lead to novel genetic information in the recipient. These errors can come in a variety of forms, from deletions to duplications, and can come on either a large or a small scale. If the change is not lethal to the organism, then there is a chance that it will become frequent in the population through random mating, a process known as genetic drift. If the change is either beneficial or detrimental, it may in fact be selected for through evolution. When such a mutation becomes frequent enough in the population, usually greater than 1% of the population, it is said to become a polymorphism. At a polymorphic loci, the different DNA variants are referred to either as polymorphisms or alleles.

When an allele has a biological effect, the terms used to describe its inheritance depend on its effect on phenotype. Loci for which two variant alleles are present can have the alleles described as being dominant, recessive or additive, or express some other non-standard form of inheritance. Of the standard forms, additive alleles are those for which each additional copy of the allele modifies the phenotype in a linear manner; an example would be allelic variant affecting the promoter of a gene involved in height, where having 0, 1 or 2 copies of one the alleles makes you proportionately that much taller. Dominant and recessive alleles are those which require one or two

alleles for the phenotype to be present, respectfully. For example, suppose there were an enzyme involved in a biological pathway which had a wild-type allele and a mutant inactive allele. If having only one copy of the wild-type allele was sufficient to maintain function, it would be said to be dominant, as its phenotype is expressed whenever it is present regardless of the other allele. By contrast, the mutant allele would be said to be recessive, as the defective phenotype would only be present if both copies of the mutated allele were present. In this case, the wild-type allele would be dominant to the recessive mutant allele.

The most common polymorphism occurs when a single base pair of DNA is replaced by another, such as when a C on one strand is replaced by a T, with the similar change in complementary base pair on the opposite strand. This sort of polymorphism is termed a Single Nucleotide Polymorphism (SNP), and each of the two bases which can be found at that location are the alleles. Of the two alleles for a SNP, the one that is observed more frequently in the population is termed the major allele, while the less frequent variant is the minor allele. A person can have either two copies of the major allele, being homozygous major, have one copy each of the major and minor alleles, making them heterozygous, or have two copies of the minor allele, making them homozygous for the minor allele, as a result of the duplicity of chromosomes in human somatic tissue.

The largest current study underway to map and study SNPs is the International HapMap Project, which has identified over 3 million SNPs to date across several populations [25]. SNPs also form the basis of the recent advent of DNA microarrays and Genome Wide Association (GWA) studies, and the race to discover previously unidentified genetic loci associated with various diseases.

2.2.2 DNA Microarrays

A person's genotype can refer to several different things. It can either be the sum total of the genetic material of a person or a cell, or it can refer locally to a specific chromosomal location and to what genetic variant a person has at that locus. At the locus level, determining an individual's genotype at a SNP can be accomplished in a number of different ways. If one is interested in determining a person's genotype at only a few bases, techniques such as Polymerase Chain Reaction (PCR) to amplify DNA, when used in conjunction with DNA sequencing, can be relatively quick and economical. However, when one is interested in determining genotypes at thousands of loci at once, it is far more practical to use DNA microarrays.

DNA microarrays are 'chips' with have thousands to millions of sites at which DNA can bind. At each of the sites, there are several short, approximately 25 base pair fragments of DNA representing the sequence around a particular SNP, and featuring both of the two alleles of that SNP. When using a DNA microarray, the DNA of an individual is first amplified to yield multiple additional copies of the genome, before being broken down into small fragments. These short fragments are then attached to fluorophores, molecules which absorb ultra-violet radiation at one wavelength and emit it at another. When the DNA of an individual is allowed to bind to the chip, it will bind preferentially to the chip sequence at that site to which it is most similar. For example, if a person is homozygous for a SNP, their DNA will bind to the sequence on the chip at the SNP specific site that contains the allele they are homozygous for. Fluorescence can then be used to determine to which sequence variants a person's DNA has bound, and to determine whether the individual is homozygous major, heterozygous or homozygous minor for that SNP. This is repeated at every site on the chip, and gives a genetic profile listing the individuals genotype at every site interrogated by the SNP chip.

In reality, this is an oversimplification of the facts, particularly with respect to

the determination of genotype by fluorescence intensity. One of the difficulties of genotyping by DNA microarray stems from the sequence surrounding the allele being genotyped. For the two different allelic versions of a SNP, the DNA surrounding the SNP is the same regardless of the variant. As such, 24 of the 25 bases are the same regardless of SNP, a 96% sequence identity. Since the probability a loose DNA fragment will bind to sequence on the chip is a function of the sequence identity, a fragment with a particular allele at a SNP will still bind with considerable frequency to the sequence at that site which contains the other allele. This is further complicated by the varying nature of the sequence surrounding the SNP, as the bond between the DNA bases C and G is stronger than the bond between A and T. As a result, if the sequence surrounding a SNP is rich in GC bonds, it will be more likely to bind to the opposite variant. Given the large number of SNPs on a chip, and the fact that every site on the chip has to be unique so as to bind uniquely to the genome, different mathematical models exist which generate genotypes based on fluorescence intensities. These models will be discussed in more detail in Section 2.3.1.

Once these models are set up, optimized and run, a chip will give a person's genotype at each of the SNPs under consideration with a certain measure of confidence. This acquisition of a person's genotype at a large number of SNPs serves as the basis for a genome wide association study.

2.2.3 Genome Wide Association Studies

A Genome Wide Association (GWA) study is a large study, where the variables of interest are genetic and obtained from DNA microarrays. GWAs are often of a case-control design, which, as the name suggests, is a type of study in which some sort of trait is used to split a population into two groups, those with the trait, known as cases, and those without the trait, known as controls. The controls can either be random population controls, of whom a fraction may have the case phenotype, or the controls

can themselves be screened to reduce the probability of having the case phenotype. Measured differences between the case and control populations for any variables can then be investigated as being associated with the trait under investigation, which, in the case of medical studies, is often a disease such as coronary artery disease.

Once the case and control groups have been established, blood samples are drawn from the subjects, and a DNA microarray is used to generate a genetic profile for each individual in the study. Other variables of interest, such as age, smoking or diabetes, may also be collected to be used as corrective factors later on in the analysis. For example, if the case group has a diabetes frequency of 30%, and the control group has a diabetes frequency of 20%, then one needs to correct for this difference to ensure that the positive results seen are not from loci which influence the risk of diabetes.

Simply put, the traditional analysis of a GWA looks for differences in frequency between case and control genotype frequencies at each locus on the chip to detect evidence of association with the primary outcome. This potentially means that several hundred thousand single variable tests of association will need to be used. The statistical methodology surrounding the analysis of genome wide association studies will be covered subsequently in the Statistical Analysis of Genome Wide Association Studies section, and is currently a matter subject to intense debate, particularly with respect to correcting for the number of comparisons employed.

2.3 Statistical Analysis of Genome Wide Association Studies

Although the description advanced in the previous section of genome wide association studies was rather straightforward, the reality is that a significant amount of statistical work must be done during analysis. The purpose of this section is to describe these processes with a suitable amount of rigour so as to either facilitate later derivations

in the Methods chapter, so that they may be coded, or so that these processes may be discussed in the Discussion with an understanding of their statistical foundation.

This section will be organized as follows. The first subsection will discuss genotype calling algorithms, which are mathematical methods for calling genotypes using DNA microarrays. The second subsection will discuss various quality control procedures which are used to ensure that the genetic data acquired from the DNA microarrays are of sufficient quality. The third subsection will describe the concept of linkage disequilibrium as a measure of the non-independence between adjacent regions of the genome. The fourth subsection will introduce various single variable tests of association, focusing on the derivation of the distribution of the test statistics under null and alternative hypotheses. The fifth subsection will deal briefly with some population genetics. Finally, the sixth subsection will feature a brief introduction to Bayesian inference and its use in hypothesis testing.

2.3.1 Genotype Calling Algorithms

To generate genotypes from DNA microarrays, genotype calling algorithms must be used. These algorithms are based on mathematical models which are varied and complex; therefore, only a brief overview of these methods will be given. Note that as this is an overview, a certain familiarity with statistical genetics will be assumed. As such, for someone unfamiliar with the area it may be prudent to read this subsection last among the material in Section 2.3.

This subsection will be organized into three subsubsections, with each subsubsection dedicated to a particular genotype calling algorithm. Note that all three genotype calling algorithms were used to generate the genetic data used later on in this thesis.

Dynamic Module

The Dynamic Module (DM) algorithm, developed by Xiaojun Di *et al.* at Affymetrix (Santa Clara) [15], is a relatively simple algorithm which was originally developed in parallel with the Affymetrix 100K SNP GeneChip; it is also usable on the Affymetrix 5.0 GeneChip with 500K SNPs.

Physically, each Affymetrix 500K GeneChip array is constructed so that for all SNPs on the array, there are n probe quartets consisting of short, 25 base pair sequences to which subject DNA can bind. Each member of a quartet is identical save for the base at the actual SNP of interest, with the four bases A, C, G and T being represented by the four members of the quartet. The difference between the n quartets themselves lies in their offset from the central SNP, with quartets being offset $-4, -2, -1, 0, +1, +3$, or $+4$ bases from the central SNP.

Once the subject DNA has been amplified, cut, attached to fluorophores, plated and had its fluorescence intensity recorded, the process of calling genotypes can begin. For each of the four submembers i of each quartet, a mean μ_i and variance σ_i^2 are calculated with respect to the measured fluorescence intensity, along with the number of incident events corresponding to fluorescence detection n_i, where $i = 1, 2, 3, 4$ are the four members of the quartet. The process of determining which of the three potential genotypes (homozygous major (AA), heterozygous (AB) and homozygous minor (BB)), along with a fourth 'no call' genotype, is most likely, is accomplished using a likelihood function. Consider first the likelihood function obtained when considering n_i samples drawn from $i = 1, 2, 3, 4$, the quartet members; the likelihood function for the means and variances $\{\hat{\mu}_i, \hat{\sigma}_i\}$ for model m would be

$$L(\{\hat{\mu}_i, \hat{\sigma}_i\}) = \prod_{i=1}^{4} \prod_{j=1}^{n_i} \frac{1}{\sqrt{2\pi}\hat{\sigma}_i} e^{\frac{-(x_{i,j}-\hat{\mu}_i)^2}{2\hat{\sigma}_i^2}}$$

where $x_{i,j}$ is the fluorescence level of the j^{th} measured fluorescence activity from the

ith quartet. Note that the hat notation refers to the estimates of the means and variances with respect to a particular model. Also, note that a normal distribution can be assumed for signal intensity as each pixel recording fluorescence intensity would likely be struck by multiple photons at a particular site. Now, taking the log of the above and simplifying, we get that

$$L(\{\hat{\mu}_i, \hat{\sigma}_i\}) = -\frac{1}{2}\sum_{i=1}^{4}\left(n_i \log 2\pi\hat{\sigma}_i^2 + \frac{\sum_{j=1}^{n_i}(x_{i,j} - \hat{\mu}_i)^2}{\hat{\sigma}_i^2}\right) \tag{2.3.1}$$

$$= -\frac{1}{2}\sum_{i=1}^{4} n_i\left(\log 2\pi\hat{\sigma}_i^2 + \frac{\sigma_i^2 + (\mu_i - \hat{\mu}_i)^2}{\hat{\sigma}_i^2}\right) \tag{2.3.2}$$

Note that the above (2.3.2) is the same as given in the original Di paper [15].

Now, with the above modified likelihood function, the MLEs for $\{\hat{\mu}_i, \hat{\sigma}_i\}$ can be calculated for each of the four models corresponding to the three potential genotypes and the no call genotype. As an example, consider the likelihood of the AA genotype, and without loss of generality assume that the A signal is coming from quartet $i = 1$. Then the likelihood is calculated under the assumption that the non-A quartets, $i = 2, 3, 4$, have means and variances which are equal and distinct from the A quartet; in other words, $\hat{\mu}_2 = \hat{\mu}_3 = \hat{\mu}_4$ and $\hat{\sigma}_2 = \hat{\sigma}_3 = \hat{\sigma}_4$ are set as being true. The maximum likelihood estimates of the means and variances $\{\hat{\mu}_i, \hat{\sigma}_i\}$ are then calculated by differentiating (2.3.2), which gives the MLEs of $\{\hat{\mu}_i, \hat{\sigma}_i\}$ as well as the maximum likelihood, $L(m)$, for the model corresponding to genotype AA.

Once $L(m)$ has been calculated for each of the models, define $S_k(m) = L_k(m) - \max\{L_k(i), i = 1, 2, 3, 4, i \neq m\}$ for each of the n different probe quartets k. Note that this is arranged so that $S_k(m) > 0$ iff it is the greatest $L(m)$ for k, $S_k(m) < 0$ iff not. Next, for each model, calculate $V(m) = \{S_1(m), S_2(m), \ldots, S_n(m)\}, m = 1, 2, 3, 4$. This leads to the notion that a particular genotype is favoured if $V(m)$ is filled with mostly positive values. To test which of the $V(m)$ are most likely to contain positive values, a Wilcoxon signed rank test can be used under the null hypothesis

H_0 : median$(S_i(m)) = 0$ vs. the alternative hypothesis H_1 : median$(S_i(m)) > 0$ to generate four p-values, p_m for $m = 1, 2, 3, 4$. The accepted genotype is then the one which has the lowest p_m as long as it meets some pre-specified threshold $p_{threshold}$. If none of the p-values are less than this threshold, or if the 'no call' option is the lowest, then the locus is not called for a genotype.

Bayesian Robust Linear Model with Mahalanobis distance classifier

Following the introduction and use of the DM genotype calling algorithm, Affymetrix, the company which manufactures GeneChips, decided to introduce their own calling algorithm. The result, the Bayesian Robust Linear Model with Mahalanobis distance classifier (BRLMM, pronounced Bee-Realm), uses the DM algorithm just described [15] while also drawing heavily from the Robust Linear Model with Mahalanobis distance classifier (RLMM), an algorithm published by Rabbee and Speed [39]. The following description of the algorithm stems from a company 'white paper' published online by Affymetrix in 2006 [1].

To begin, the raw probe intensities are normalized and are used to generate allele signal estimates S_A and S_B for each of the two alleles A and B for each of the SNPs on the chip for each person being genotyped. The derivation of these estimates is described vaguely by the authors as "We therefore summarize these intensities in a single value for the features corresponding to each allele, the 'signal' for the allele" [1]; and are said to be such that the A signal is proportional to the amount of A DNA present, while the B signal is proportional to the amount of B DNA present.

Next, a cluster space transformation is made to the data for each SNP, which is currently an $2 \times N$ matrix of allele signal estimates. The transformation is in several parts, beginning by a transformation to two new measures, the 'signal strength' $\log_2(S_A) + \log_2(S_B)$ and the 'allele constrast' $\frac{S_A - S_B}{S_A + S_B}$. These measures are further transformed by a 'cluster-center-stretch' transformation which accounts for a discrep-

ancy in heterozygote versus homozygote variance in transformed allele signal estimate cluster variance. At this point the data can be represented visually by plotting the signal strength against the allele constrast; the result is a set of three clusters, which are, generally speaking, homozygote BB to the left, heterozygote AB in the middle, and homozygote AA to the right. The next step of BRLMM is to try and identify cluster centers and variances accurately so that any sample can be called by analyzing which cluster it is most likely to belong to.

Estimating cluster center and variance is done in a two part fashion. First, an initial estimate of cluster center and variance is obtained from a subset of all SNPs. This is accomplished by running the DM algorithm with what the authors describe as "a highly-stringent confidence threshold of 0.17 to determine initial genotype calls" [1]. Note that no further derivation, results or explanation are given for this threshold. Next, for SNPs with a certain threshold number of DM calls made for all three genotypes, a subset of 5000 SNPs is drawn and a representative cluster center and variance are taken, which becomes the prior cluster center and variance used later on for all SNPs.

Next, for each SNP, using only those subjects which have been called by the stringent DM, a locus specific cluster center and variance is taken for the 3 genotypes, if there are DM calls for that cluster. Next, a posterior estimate of cluster center and variance is taken by taking a weighted combination of the prior estimate of cluster center and variance and updating it with the SNP specific information. It is this center and variance which is used to make genotype calls, with the help of the Mahalanobis distance

$$d(x, \mu, \Sigma) = \sqrt{(x-\mu)^t \Sigma^{-1} (x-\mu)}$$

where x is the test value, μ is the cluster center and Σ the covariance matrix. Note that x and μ are two dimensional, Σ is 2×2. Finally, for each of the three μ's, a distance is obtained. The genotype call is the one with the lowest distance, call it d_1.

Additionally, a measure of confidence is obtained by dividing d_1 by d_2. This measure, $\frac{d_1}{d_2}$, which falls between 0 and 1, can then be compared against a threshold, say $p_{threshold} = 0.5$, and genotype calls made only if the threshold is met. This measure of confidence, much like with DM, can be compared against a threshold, which sets up a trade-off of call rate vs. accuracy; lowering the threshold increases the call rate and decreases the accuracy, and vice-versa. The recommended $p_{threshold}$ of 0.5 is again given without a particular statistical foundation but comes from empirical results.

Birdseed

The Birdseed genotype calling algorithm was published in 2008 in *Nature Genetics* by J Korn *et al.* [28]. Birdseed is part of a larger suite of programs, appropriately titled Birdsuite, which act to analyze results from the Affymetrix 1M Genechip, the successor to the Affymetrix 500K Genechip.

Birdseed, much like BRLMM, starts by normalizing the raw fluorescence intensities. Once appropriately normalized, the fluorescence intensities take the form of a $2 \times N$ matrix for each of the SNPs on the array, where one value represents the signal of the A allele and the other the B allele. Birdseed works by taking these raw fluorescence intensities and uses a supervised learning technique to cluster the samples together into sets which should have the same genotype.

The clustering used by Birdseed attempts to fit the data to a 2-dimensional Gaussian Mixture Model based on either $1, 2$ or 3-clusters, depending on how many of the three genotypes are present. Before it begins, it draws an estimate of cluster centers and variances from a separate input file. This file contains SNP specific cluster centers and variances obtained from genotyping HapMap samples on the Affymetrix 1M platform, which are assumed to be accurate due to high concordance with other genotyping platforms which have also been applied to HapMap samples. Next, an Expectation-Maximization iterative process is applied, so that on each iteration the

following two steps occur: the current cluster centers and variances are used to infer genotypes of the N samples; the genotypes are used to generate new cluster centers and variances. This iterative process is run until convergence, whereby class membership, and hence genotype, is inferred from the class j which, for a sample i, maximizes the equation

$$P(j|i) = K\frac{w_j}{|\Sigma_j|^{\frac{1}{2}}}e^{\frac{1}{2}(x_i-\mu_j)^T\Sigma_j^{-1}(x_i-\mu_j)}$$

where μ_j is the mean for cluster j, Σ_j is the variance for cluster j, w_j is the 'weight' for cluster j, defined to be a third for each cluster in a 3-cluster model and a half for each cluster included in a 2-cluster model, and K is a normalization constant so that $\sum_j P(j|i) = 1$. Note that the above equation is similar to the pdf of a multivariate normal distribution, as the underlying Gaussian Mixture Model assumes the data is drawn from clusters which have a multivariate normal distribution.

Finally, a measure of confidence is assessed for each genotype for each SNP, depending on a weighted sum of the number of standard deviations away a sample is from is most likely cluster, and the ratio of $\frac{P(j_{2nd_{best}}|i)}{P(j_{best}|i)}$. This, much like in DM and BRLMM, allows the fine tuning of the ratio between call rate and accuracy in the calling of genotypes.

2.3.2 Quality Control

Quality Control (QC) is the process of ensuring that the data one uses for analysis is of an acceptable quality. Particularly with a genetic study, where there are copious amounts of data, QC is the process performed before any statistical analysis which removes results which are likely to be inaccurate due to illogical genotype distributions, problematic distributions which are frequently inaccurately called by genotype calling algorithms or individuals which have outlier genotype distributions. Although QC is a multi-layered process, the majority of the steps involved are rather straightforward, and only two require significant explanation. The purpose of this subsection

is to explain those two steps.

This subsection will be organized as follows. The first subsubsection will deal with the removal of subjects who are deemed to be genetically different from the main body of subjects of a genome wide association study by the use of Principal Components Analysis (PCA). Finally, the second subsubsection will deal with the removal of SNPs for which the control genotypes are not in Hardy-Weinberg Equilibrium, a model which describes the expected genotype frequencies in the population based on the allele frequencies.

Principal Components Analysis

Determining the principal components of a group of variables can serve several purposes. It has often been used for modeling purposes in analyses such as logistic regression where there is a desire to replace a set of related variables by a smaller number of variables which explain a large proportion of the variance of the initial set of variables [19]. For example, if there were a set of variables $X_1, \ldots, X_m$, Principal Components Analysis (PCA) allows for the determination of eigenvectors $\mathbf{e_1}, \ldots, \mathbf{e_m}$ from the covariance matrix between X_i's, with the eigenvectors being ordered based on their ability to successively account for as much of the variance in the original set of variables as possible. As such, instead of including variables $X_1, \ldots, X_m$ in the model, it may be possible to include a new set of variables $X'_1, ..., X'_p$ where $X'_i = \mathbf{e_i}^{\mathrm{T}}(X_1, \ldots, X_m)$ and with $X'_1, \ldots, X'_p$ explaining a certain desired proportion of the variance in $X_1, \ldots, X_m$, say 80%.

Consider the above in a more formal manner. We are interested in drawing iid m-dimensional random variables $\mathbf{X} = (X_1, \ldots, X_m)$, which have some expected value $\mu = (\mathrm{E}(X_1), \ldots, \mathrm{E}(X_m))$, and some covariance matrix $\Sigma = \mathrm{Cov}(\mathbf{X}, \mathbf{X})$. For a particular sample of n $\mathbf{X}$'s, $\mathbf{x_1}, \ldots, \mathbf{x_n}$, we can calculate the sample mean $\overline{\mathbf{x}} = (\sum_{i=1}^{n} \mathbf{x_{1,i}}, \ldots, \sum_{i=1}^{n} \mathbf{x_{m,i}})$ and the sample covariance matrix S with entries $S_{i,j} =$

$\frac{1}{n-1}\sum_{k=1}^{n}(\mathbf{x_{i,k}}-\overline{\mathbf{x_i}})(\mathbf{x_{j,k}}-\overline{\mathbf{x_j}})$ which are unbiased estimators of the expected value and covariance matrix, respectively. Consider the following theorem, which is paraphrased from Applied Multivariate Statistical Analysis (p373, [26]).

Theorem 2.3.1 Sample Principal Components *For a sample covariance matrix S with eigenvalue-eigenvector pairs $(\hat{\lambda}_1, \hat{\mathbf{e}}_\mathbf{1}), \ldots, (\hat{\lambda}_m, \hat{\mathbf{e}}_\mathbf{m})$, the ith sample principal component is given by $\hat{y}_i = \hat{\mathbf{e}}_\mathbf{i}^{\mathrm{T}}\mathbf{x}$. The sample variances of the sample principal components are given by* SampleVariance $(\hat{y}_i) = \hat{\lambda}_i, 1 \leq i \leq m$. *The total sample variance is given by* $\sum_{i=1}^{n} S_{i,i} = \hat{\lambda}_1 + \cdots \hat{\lambda}_m$.

Proof: For a proof of the above, please see [26]. ∎

One can use the above to explain a certain desired proportion of the sample variance by selecting an appropriate number of $\hat{\lambda}$'s. For example, if one wanted to represent 80% of the variance, one could select the smallest k such that $\frac{\sum_{i=1}^{k}\hat{\lambda}_i}{\sum_{i=1}^{m}\hat{\lambda}_i} > 0.80$.

Now, let us consider the application of PCA to DNA microarray data. Unlike the form of PCA illustrated above, we will use PCA as a means to identify people who are outliers with respect to their first few eigenvectors of the sample covariance matrix. The method we will use comes from the Reich lab at Harvard, and is explained in two papers: the earlier paper deals largely with the method as we shall use it [36], while the second paper deals more with additional theoretical considerations, such as the calculation of the cumulative distribution function of the largest eigenvalue generated by PCA, sample size considerations with being able to identify people from different populations, and calculations of principal components when markers are related by linkage disequilibrium [34].

To begin, consider the following nomenclature, which has been partially synchronized with Section 2.4, the section on machine learning, as well as the Methods chapter, and whenever possible, the original papers themselves [36, 34] and the above

explanation of the traditional use of PCA. Let i be an index for a person's genotype under consideration and let there be n genotypes under consideration, so that $1 \leq i \leq n$. Let a person's genotype $\mathbf{x_i}$ be drawn from $\mathcal{X}$, which is an m dimensional space of genotypes where m is the number of SNPs; in other words, $\mathcal{X} = \{0, 1, 2\}^m$. A person i will therefore have genotype $\mathbf{x_{i,j}}$ at SNP j. Now consider a matrix G such that $[G]_{i,j} = \mathbf{x_{i,j}}$. G is a large matrix of dimension $n \times m$ where each subject represents a row and each column a SNP.

Now, the use of PCA that we employ, based on the method of the Reich lab at Harvard [36, 34] uses a different sample covariance matrix than one might expect based on the earlier explanation. Normally, one may consider trying to create a sample covariance matrix between variables, say Q, where Q is of dimension $m \times m$. However, the version of PCA used by the Reich lab method creates a sample covariance matrix *between individuals*, a matrix Ψ, of dimension $n \times n$.

In their papers, this point of view is largely ignored. One interpretation of the matrix Ψ as opposed to the usual Q is that Ψ would be that the sample covariance matrix that you would create if the random variable under analysis was the sampling itself of n people, each drawn with respect to some underlying population. That is, you are trying to form a sample covariance matrix to approximate the relatedness that you would expect in forming repeated samplings of n people, where each person i in the drawing of n people is always drawn with respect to some underlying population P_i for that individual. One could now imagine that each of the SNPs represents a sampling, so that the sample of n people is drawn m times, which would allow for the creation of the appropriate sample covariance matrix. To do this, however, the SNPs would need to have equal expected values and variances. This is accomplished by normalizing each of the SNPs individually before the sample covariance between individuals is taken.

Consider the above in more detail. We need to normalize the information provided by each of the SNPs by transforming the matrix G into a form where the SNPs

have a more appropriate expected value and variance. Consider first the creation of the matrix G' from G by subtracting the column means $\mu_j = \frac{1}{n}\sum_{i=1}^{n} G_{i,j}$, that is $G'_{i,j} = G_{i,j} - \mu_j$. Next, consider a normalization procedure, where each of the columns are divided by $\sqrt{p_j(1-p_j)}$, where p_j is the posterior estimate of allele frequency for SNP j. That is, let

$$G''_{i,j} = \frac{G'_{i,j}}{\sqrt{p_j(1-p_j)}} = \frac{G_{i,j} - \mu_j}{\sqrt{p_j(1-p_j)}} \tag{2.3.3}$$

where $p_j = \frac{\mu_j}{2}$.

Now, the creation of the sample covariance matrix from G'' can be interpreted in two different ways. If we do not assume that the population is in Hardy-Weinberg Equilibrium, then the above is merely a procedure which the authors say "is motivated by the fact that the frequency change of a SNP due to genetic drift occurs at a rate proportional to $\sqrt{p_j(1-p_j)}$ per generation" [34]. However, if we do assume that the SNP is in Hardy-Weinberg Equilibrium, then the above normalization has a statistical meaning. Note that the authors mention both the above normalization and the following statistical motivation which follow in their paper [34]. Note further that a more detailed explanation of Hardy-Weinberg Equilibrium follows in the folowing subsubsection.

Consider the distribution of genotypes for a particular SNP j. If we assume that the SNP is in Hardy-Weinberg Equilibrium with a minor-allele frequency of p, and independent of all other SNPs under consideration, then for person i the genotype $\mathbf{X_{i,j}}$ will be drawn from a Binomial distribution with $n = 2$, or in other words

$$f(\mathbf{X_{i,j}} = x|n,p) = \binom{2}{x} p^x (1-p)^{2-x}$$

with $\mathrm{E}(\mathbf{X_j}) = 2p$ and $\mathrm{Var}(\mathbf{X_j}) = 2p(1-p)$. Now, the sample mean is an unbiased estimator for $2p$, so the sample mean divided by 2 is an unbiased estimator for p, in other words, $p_j = \frac{\mu_j}{2} = \frac{1}{2n}\sum_{i=1}^{n} \mathbf{x_{i,j}}$ is an unbiased estimator for p. Similarly, the

expected value of $p_j(1-p_j)$ is

$$\begin{aligned}
\mathrm{E}(p_j(1-p_j)) &= \mathrm{E}\left((\frac{1}{2n}\sum_{i=1}^{n}\mathbf{X_{i,j}})(1-\frac{1}{2n}\sum_{i=1}^{n}\mathbf{X_{i,j}})\right) \\
&= \frac{1}{2n}n\,\mathrm{E}(\mathbf{X_j}) - \frac{1}{4n^2}\,\mathrm{E}\left((\sum_{i=1}^{n}\mathbf{X_{i,j}})(\sum_{i=1}^{n}\mathbf{X_{i,j}})\right) \\
&= p - \frac{1}{4n^2}n\,\mathrm{E}(\mathbf{X_j}^2) - \frac{1}{4n^2}n(n-1)(\mathrm{E}(\mathbf{X_j}))^2 \\
&= p - \frac{p^2}{2n} - \frac{p}{2n} - p^2 + \frac{p^2}{n} \\
&= \left(\frac{2n-1}{2n}\right)p(1-p)
\end{aligned}$$

and so $\sqrt{p_j(1-p_j)}$ is a biased estimator for $\sqrt{\frac{1}{2}\mathrm{Var(X)}}$. At the very least, it normalizes each of the SNPs such that they have about the same variance, 2.

Therefore, the SNPs in normalized matrix G'' from equation (2.3.3) can be assumed to have expected value 0 since their means μ_j have been subtracted, and additionally they can be assumed to have either equal variance almost equal to about 2 or have been normalized relative to some level of genetic drift. This allows us to calculate a sample covariance matrix by taking as the sample covariance between individuals i and i' to be the covariance between the SNPs of those two people; in other words, $\Psi_{i,i'} = \mathrm{Cov}(\mathbf{x_i}, \mathbf{x_{i'}})$. It turns out that the entire matrix Ψ can be calculated relatively easily by the matrix calculation

$$\Psi = (G'')(G'')^{\mathrm{T}}$$

The eigenvectors and eigenvalues of Ψ can be calculated relatively easily using techniques such as singular value decomposition. Price *et al.* define outliers by first projecting each individual onto their top 10 eigenvectors, and then removing any individual greater than 6 standard deviations away from the mean along a particular eigenvector [36]. Note that no particular rationale is given for this step in the paper.

This last step is applied iteratively to successively remove individuals of decreasing heterogeneity until a desired level of population homogeneity is acquired.

Hardy-Weinberg Equilibrium

In essence, the principle of Hardy-Weinberg Equilibrium (HWE) states that for a locus with two variants, such as a major allele A and minor allele a, the expected genotype frequencies in a population can be determined by the major and minor allele frequencies in the population. That is, given a population allele frequency for A of p_A, and for a of $p_a = 1 - p_A$, the population genotype frequencies are expected to be $p_{AA} = p_A^2$, $p_{Aa} = 2p_Ap_a$ and $p_{aa} = p_a^2$. Note that HWE describes what happens under an ideal setting, where the following assumptions hold true: that neither A nor a preferentially affects the survival or reproductive capabilities of the organism; that the population size is infinite; that no genetic drift is occurring; and that mating is done at random. These assumptions are, of course, impossible in nature, particularly the infinite population size, but for control populations in genetics studies the principle of Hardy-Weinberg Equilibrium gives a rough estimate as to the relationship between genotype and allele frequencies.

With respect to genetics studies such as GWAs, determining if SNPs are in HWE is useful in that it can help to uncover potential genotyping errors. If a locus is in HWE, which we assume every SNP in a GWA is for the control population, then the calculated genotype and allele frequencies will be related and have related distributions. If they are not, it is usually a sign that the genotyping process undertaken by the GeneChip is inaccurate, and that the results for that SNP are not of sufficient quality. To test whether HWE holds for a given SNP, an Exact test is performed in the following manner. Note that the derivation which follows is based around work presented in Genetic Data Analysis II by Weir [50].

Consider first the following two probability distributions. The first is the multi-

nomial genotype probability of seeing a set of genotype counts n_{AA}, n_{Aa} and n_{aa} given an underlying allele frequency p_A

$$f(n_{AA}, n_{Aa}, n_{aa}|p_A) = \frac{n!}{n_{AA}!n_{Aa}!n_{Aa}!}(p_A^2)^{n_{AA}}(2p_Ap_a)^{n_{Aa}}(p_a^2)^{n_{aa}} \tag{2.3.4}$$

Now consider the binomial allele probability distribution; that is, given p_A and a given sample size $n = \frac{1}{2}(n_A + n_a)$, the probability of seeing n_A major alleles and n_a minors alleles is

$$f(n_A, n_a|p_A) = \frac{(2n)!}{n_A!n_a!}(p_A)^{n_A}(p_a)^{n_a} \tag{2.3.5}$$

Both these probability distributions are conditional on the underlying population major allele frequency p_A. However, it turns out that if we condition (2.3.4) by (2.3.5), then we obtain a distribution unencumbered by p_A. Note that the following makes use of the fact that $n_A = 2n_{AA} + n_{Aa}$ and $n_a = 2n_{aa} + n_{Aa}$.

$$\begin{aligned} f(n_{AA}, n_{Aa}, n_{aa}|n_A, n_a) &= \frac{f(n_{AA}, n_{Aa}, n_{aa}, n_A, n_a|p_A)}{f(n_A, n_a|p_A)} \\ &= \frac{f(n_{AA}, n_{Aa}, n_{aa}|p_A)}{f(n_A, n_a|p_A)} \\ &= \frac{n!}{n_{AA}!n_{Aa}!n_{Aa}!}\frac{n_A!n_a!}{(2n)!}(p_A)^{2n_{AA}+n_{Aa}-n_A}(p_a)^{2n_{aa}+n_{Aa}-n_a}2^{n_{Aa}} \\ &= \frac{2^{n_{Aa}}n!n_A!n_a!}{n_{AA}!n_{Aa}!n_{Aa}!(2n)!} \end{aligned} \tag{2.3.6}$$

Furthermore, if we have n_A and n_a, as we are assumed to have in the conditional probability of (2.3.6) above, then the distributions of n_{AA}, n_{Aa} and n_{aa} will be related. Namely, we can rewrite the homozygous counts, n_{AA} and n_{aa} if we know n_A, n_a and n_{Aa} as $n_{AA} = \frac{1}{2}(n_A - n_{Aa})$ and $n_{aa} = \frac{1}{2}(n_a - n_{Aa})$, which allows us to rewrite (2.3.6) as the equivalent distribution

$$f(n_{Aa}|n_A, n_a) = \frac{2^{n_{Aa}}(n)!n_A!n_a!}{(\frac{1}{2}(n_A - n_{Aa}))!n_{Aa}!(\frac{1}{2}(n_a - n_{Aa}))!(2n)!} \tag{2.3.7}$$

The above equation (2.3.7) represents the probability of seeing a given number of heterozygotes given an observed number of minor and major alleles. To calculate a p-value, one first generates the probability distribution function by calculating a probability for every possible heterozygote combination given n_A and n_a. The p-value for an observed number of heterozygotes is then taken as the sum of the probabilities which are lower than the probability for the observed number of heterozygotes, which makes it an Exact test for determining whether or not a locus is in Hardy-Weinberg equilibrium.

2.3.3 Linkage Disequilibrium

During meiosis and reproduction in diploid species such as humans, half of the chromosomes come from one parent and half from the other. Ignoring population level genetics, any genetic material that comes from one parent will be inherited independently of the other parent, as the chromosomes assort into gametes in the respective parent. However, within the genetic material provided by one parent, linkage may occur between two genetic loci on the same chromosome. That is, while loci on different chromosomes are always inherited independently of each other, loci which are on the same chromosome may be inherited in some sort of non-independent fashion, depending on how far apart on the chromosome they are. The purpose of this subsection is to describe a common method for quantifying the level of non-independence between loci at a population level, a topic known as linkage disequilibrium.

This subsection will be organized as follows. The first subsubsection will introduce the problem and the relevant nomenclature. The second subsubsection will discuss the process of estimating two-locus haplotype frequencies, with a haplotype being a block of conserved DNA. Finally, the third subsubsection will describe how the popular r^2 and D' measures of linkage disequilibrium can be calculated from two-locus haplotype frequencies.

Introduction

Linkage Disequilibrium (LD) describes the non-random assortment of genotypes. There are various ways by which one can measure linkage disequilibrium, but they are all based on the underlying notion of a haplotype. In short, a haplotype is a continuous sequence of DNA in which recombination during meiosis is unlikely to occur, so that DNA is normally inherited together in a block. These haplotype blocks vary in length, but are often several hundred thousand bases long.

Consider two nearby SNPs in the genome; SNP 1 with genotype A vs a, and SNP 2, with genotype B vs b. From those two SNPs, four potential haplotypes can arise: A and B; A and b; a and B; and a and b. The measures of linkage disequilibrium we are concerned with will ask the question of whether the probability of seeing A and B together based on the underlying haploptypes, p_{AB}, is different from the product of their underlying single SNP genotype frequencies, $p_A \times p_B$. However, before we can estimate linkage disequilibrium from haplotype frequencies, we need to generate haplotype frequencies from the observed genotype contingency tables.

Note that the following discussion of the generation of two-locus haplotype frequencies, and the subsequent calculation of linkage disequilibrium measures, largely stems from the book Genetic Data Analysis II by Weir [50].

Estimating Two Locus Haplotype Frequencies

Estimating two locus haplotype frequencies is a process which is surprisingly more difficult than one might expect. Consider the acquisition of genotype information; that is, two loci are genotyped independently of each other in a sample population. The results will look like those shown in Table 2.1. Using these results, we can generate the two locus genotype frequencies; for example, one can obtain the frequency of samples which are homozygous for both genotypes by dividing by the number of samples, or $p_{AABB} = \frac{n_{AABB}}{n}$. To obtain the related but different haplotype frequencies

is more complicated, as we can only observe genotype, not haplotype frequencies.

	SNP1 (A vs. a)		
SNP2 (B vs. b)	n_{AABB}	n_{AaBB}	n_{aaBB}
	n_{AABb}	n_{AaBb}	n_{aaBb}
	n_{AAbb}	n_{Aabb}	n_{aabb}

Table 2.1: Nomenclature for Contingency Table with Two SNPs

For most genotype frequencies, there is a direct one to one relation with haplotype frequencies; for example, the genotype combination $AABb$ is caused only by the underlying haplotypes AB and Ab. The problem lies with the genotyped double heterozygote, $AaBb$, which may be caused by haplotype combinations AB and ab or Ab and aB. This means we cannot directly calculate haplotype frequency; for example p_{AB} becomes

$$\begin{aligned} p_{AB} &= p_{AB}^{AB} + \frac{1}{2}\left(p_{Ab}^{AB} + p_{aB}^{AB} + p_{ab}^{AB}\right) \\ &= p_{AABB} + \frac{1}{2}\left(p_{AaBB} + p_{AABb} + p_{ab}^{AB}\right) \end{aligned} \tag{2.3.8}$$

which we cannot directly calculate because we cannot disentangle the double heterozygote $p_{AaBb} = p_{ab}^{AB} + p_{Ab}^{aB}$. However, provided certain assumptions, such as the assumption that the population is in Hardy-Weinberg Equilibrium, we can determine what is the most likely frequency for the AB haplotype by breaking down p_{AaBb}. To do this, consider first the frequencies of the other two locus haplotypes. They are

$$\begin{aligned} p_{Ab} &= p_A - p_{AB} \\ p_{aB} &= p_B - p_{AB} \\ p_{ab} &= 1 - p_A - p_B + p_{AB} \end{aligned}$$

Therefore, under HWE and with the given haplotype frequencies, we can estimate

that the genotyped double heterozygote is likely to be composed as follows

$$p_{AaBb} = p_{ab}^{AB} + p_{Ab}^{aB} = \left(\frac{p_{AB}p_{ab}}{p_{AB}p_{ab} + p_{Ab}p_{aB}}\right) p_{AaBb} + \left(\frac{p_{Ab}p_{aB}}{p_{AB}p_{ab} + p_{Ab}p_{aB}}\right) p_{AaBb}$$

As a result, the earlier probability of p_{AB} given in equation (2.3.8) can be rewritten as

$$p_{AB} = p_{AABB} + \frac{1}{2}\left(p_{AaBB} + p_{AABb} + \left(\frac{p_{AB}p_{ab}}{p_{AB}p_{ab} + p_{Ab}p_{aB}}\right) p_{AaBb}\right) \tag{2.3.9}$$

The above is actually a cubic equation in terms of p_{AB}, and as such can be solved using a polynomial root solver to yield three potential values for p_{AB}. Determining which of those is the most likely value for p_{AB} is done using a maximum likelihood estimator in the following manner. Consider the multinomial haplotype distribution that would be present based on the haplotype frequencies; the probability of obtaining a given result would be as follows

$$f(n_{AB}, n_{Ab}, n_{aB}, n_{ab}; p_{AB}, p_{Ab}, p_{aB}, p_{ab})$$

$$= \frac{n!}{n_{AB}!n_{Ab}!n_{aB}!n_{ab}!}(p_{AB})^{n_{AB}}(p_{Ab})^{n_{Ab}}(p_{aB})^{n_{aB}}(p_{ab})^{n_{ab}}$$

This in turns yields a simplified version of the likelihood function for the parameter p_{AB}, which is

$$L(p_{AB})$$
$$= \frac{n!}{n_{AB}!n_{Ab}!n_{aB}!n_{ab}!}(p_{AB})^{n_{AB}}(p_A - p_{AB})^{n_{Ab}}(p_B - p_{AB})^{n_{aB}}(1 - pA - pB + p_{AB})^{n_{ab}}$$

If we rewrite the above to use the observed genotype counts and take its logarithm,

we get

$$\begin{aligned}\log(L(p_{AB})) = Constant+ \\ (2n_{AABB} + n_{AaBB} + n_{AABb})\log(p_{AB})+ \\ (2n_{AAbb} + n_{AABb} + n_{Aabb})\log(p_A - p_{Ab})+ \\ (2n_{aaBB} + n_{AaBB} + n_{aaBb})\log(p_B - p_{aB})+ \\ (2n_{aabb} + n_{Aabb} + n_{aaBb})\log(1 - p_A - p_B + p_{ab})+ \\ n_{AaBb}\log\left(p_{AB}(1 - p_A - p_B + p_{ab}) + (p_A - p_{aB})(p_B - p_{aB})\right) \qquad (2.3.10)\end{aligned}$$

Using the three potential values for p_{AB} given by equation (2.3.9), we can calculate the logarithm of the likelihood function for each value. However, some of the values for p_{AB} from (2.3.9) may not be acceptable as probabilities, as the value for p_{AB} must be such that the other haplotype probabilities, $p_{Ab} = p_a - p_{AB}$, $p_{aB} = p_B - p_{AB}$ and $p_{ab} = 1 - p_A - p_B + p_{AB}$ must all be between 0 and 1. The value of p_{AB} is therefore taken as the value which maximizes the likelihood function under the constraint that the other haplotype frequencies, p_{Ab}, p_{aB} and p_{ab}, are all between 0 and 1. Using p_{AB} and the other two locus haplotype frequencies, we are then free to calculate the various linkage disequilibrium measures in the following section.

Linkage Disequilibrium Measures

Of the multitude of linkage disequilibrium measures that exist, the most commonly used in genome wide association studies are the r^2 and D' measures. They measure similar but related phenomena, with r^2 being used primarily as a measure of how well one SNP serves as a surrogate for the other, while D' is used to detect the level of recombination between two SNPs.

Both r^2 and D' have the same numerator, the linkage disequilibrium coefficient $D = p_{AB} - p_A p_B$, which measures the deviation between the haplotype frequency

p_{AB} from that expected by the individual SNP frequencies p_A and p_B. The difference between them lies with their denominator, as listed below. Note that the formulae for the two measures was acquired from documentation for the SAS statistical software package [42].

$$D' = \left| \frac{D}{D_{max}} \right|, \quad D_{max} = \begin{cases} \min(p_A(1-p_B), (1-p_A)p_B), & D > 0 \\ \min(p_A p_B, (1-p_A)(1-p_B)), & D < 0 \end{cases}$$

$$r^2 = \left(\frac{D}{\sqrt{p_A p_B (1-p_A)(1-p_B)}} \right)^2$$

It is worth noting that $0 \leq r^2 \leq D' \leq 1$. That being said, the denominators of the two measures are used to scale the linkage disequilibrium coefficient as follows. If the two SNPs serve as perfect surrogates for each other, that is, there is a one-to-one relation between having a SNP at one locus and having a SNP at the other locus, then $p_A = p_B = p_{AB}$ and so $r^2 = 1$. However, if the two SNPs have a different frequency, $p_A \neq p_B$, but there is no recombination between them, then you can still have $p_A = p_{AB}$ or $p_B = p_{AB}$, in which case $D' = 1$. Therefore; an r^2 of 1 indicates that the two SNPs are perfectly linked and can be used as surrogates, while a D' of 1 indicates that there is no recombination between them and that haplotype analysis may be useful to understand the underlying genetic inheritance.

2.3.4 Single Variable Tests of Association

Once the genotypes have been called, and quality control measures are used to ensure accuracy of the data, it is time to associate probabilities to the results generated from the DNA microarrays. The purpose of this subsection is to introduce three single variable tests of association which are commonly used in genome wide association studies. Note that several of these tests are derived with a fair amount of detail as an understanding of them is for the later derivations in the Methods chapter; particularly

with respect to the issue of calculating power under alternative hypotheses.

This subsubsection will be organized as follows. The first subsubsection will introduce some relevant nomenclature. The next three subsubsections will deal with three single variable tests of association: Fisher's Exact Test; the Pearson Chi-Square test; and the the Cochran-Armitage Test for Trend, respectively.

Introduction

To summarize the results of an individual SNP in a GWA study, a 2×3 contingency table can be created for each of the SNPs in the study, as shown in Table 2.2. For any entry $n_{i,j}$ in the table, i is the outcome of control (0) versus CAD (1), while j is the genotype of homozygous major (0), heterozygous (1) or homozygous minor (2).

	AA	Aa	aa	
Control	$n_{0,0}$	$n_{0,1}$	$n_{0,2}$	$n_{0,+}$
Case	$n_{1,0}$	$n_{1,1}$	$n_{1,2}$	$n_{1,+}$
	$n_{+,0}$	$n_{+,1}$	$n_{+,2}$	n

Table 2.2: Nomenclature for Contingency Table with One SNP.

Choosing which statistical test to use depends on assumptions made about the underlying inheritance patterns of the SNPs in the study. Fisher's Exact Test works under the null hypothesis that the probability of any genotype is the same between case and control. The Chi-Square test works under the same null hypothesis, and is asymptotically equivalent, but has properties which make calculating the distribution of the test statistic under alternative hypotheses easier. The Cochran-Armitage Test for Trend works under the null hypothesis that there exists a linear trend with slope zero between the probabilities of being a case given a genotype across the three genotypes. There are also Dominant and Recessive versions of Fisher's Exact Test and the Chi-Square test, which work by compressing the 2×3 contingency table into a 2×2 table by grouping the heterozygotes with either the major or minor allele homozygotes, but they are not used frequently in GWAs; there use is generally

reserved for situations in which there is an *a priori* hypothesis suggesting that the locus under consideration follows that pattern of inheritance.

Fisher's Exact Test

Fisher's Exact Test works by testing to see if the distributions of the explanatory variable conditioned by the response are the same; that is, by testing to see whether the distribution of the genotype conditioned on outcome is equivalent for each genotype, $\pi_{j|0} = \pi_{j|1}$ for $j = 0, 1, 2$, where $\pi_{j|i} = P(X = j|Y = i)$, based on a contingency table for a given SNP. The eventual determination of how probable a given contingency table was depends on looking at every other contingency table with the same row and column sums, and summing up the probabilities for those tables less probable than the given one.

Unfortunately, the derivation of Fisher's Exact Test is a complicated one, and a detailed understanding of the distribution from which p-values are generated is not essential for either an understanding of the major issues of the Discussion or for programming purposes. As such, its derivation will be omitted; for a detailed derivation, please see p. 63 of Categorical Data Analysis by Agresti [2].

Pearson's Chi-Square Test

Testing for significance using Pearson's Chi-Square Test is a useful process, which is similar to Fisher's Exact Test in that in looks for the probability of observing a given contingency table given that the expected frequency of a given cell is a product of the marginals. In terms of computing p-values, given the computational abilities and relevant graph-theory algorithms available today, Fisher's Exact Test presents a superior method. However, since calculating probabilities under alternate hypotheses is more straightforward for contingency tables which are greater than 2×2, we considered the Pearson Chi-Square Test along with the Fisher's Exact Test. Since we

are considering genotypes, we will only consider contingency tables which are 2×3, following the nomenclature of Table 2.2. Note that the derivation of the distribution of the statistic under the alternative hypothesis which follows was not based on a previous work but was derived *de novo* for the purposes of this thesis.

In the 2×3 form, the Pearson Chi-Square Test works under the null hypothesis that the proportions $\pi_{j|0} = \pi_{j|1}$ for $j \in \{0, 1, 2\}$, where $\pi_{j|i} = P(X = j|Y = i)$. Consider a particular set of observed genotypes, with fixed total value n and fixed marginals $n_{i,+}$ and $n_{+,j}$. Then any particular observed proportion $P_j = \frac{n_{1,j}}{n_{+,j}}$ depends only on the distribution of $n_{1,j}$, since $n_{+,j}$ is fixed. Consider a particular column, and hence genotype. Then we get that $n_{1,j}$ is distributed as follows

$$f(n_{1,j}|n_{+,j}, n_{i,+}, n) = \frac{n_{+,j}!}{n_{1,j}!n_{0,j}!} \left(\frac{n_{1,+}}{n}\right)^{n_{1,j}} \left(\frac{n_{0,+}}{n}\right)^{n_{0,j}} \tag{2.3.11}$$

In other words, it follows a binomial distribution $\mathrm{Bin}(N, P)$ with $N = n_{+,j}$ and $P = \frac{n_{1,+}}{n}$. Therefore, the expected value and variance of P_j are given by

$$\begin{aligned} \mathrm{E}(P_j) &= \frac{1}{n_{+,j}} \mathrm{E}(n_{1,j}) = \frac{1}{n_{+,j}} n_{+,j} \left(\frac{n_{1,+}}{n}\right) = \frac{n_{1,+}}{n} \\ \mathrm{Var}(P_j) &= \left(\frac{1}{n_{+,j}}\right)^2 \mathrm{Var}(n_{1,j}) = \left(\frac{1}{n_{+,j}}\right)^2 n_{+,j} \left(\frac{n_{1,+}}{n}\right) \left(\frac{n_{0,+}}{n}\right) \\ &= \left(\frac{1}{n_{+,j}}\right) \left(\frac{n_{1,+}}{n}\right) \left(\frac{n_{0,+}}{n}\right) \end{aligned} \tag{2.3.12}$$

Therefore, by the Central Limit Theorem, each of the statistics

$$U_j = \frac{\frac{n_{1,j}}{n_{+,j}} - \frac{n_{1,+}}{n}}{\sqrt{(\frac{n_{0,+}}{n})(\frac{n_{1,+}}{n})(\frac{1}{n_{+,j}})}} \tag{2.3.13}$$

should be asymptotically distributed as a Normal $(0, 1)$ random variable.

Now, in a paper by Armitage from 1955 (where he describes the trend test that bears his name), a description is given of the Pearson Chi-Square Test whereby for a

given contingency table with 2 rows and k columns, for U_j as described above, that the sum $\sum_{i=0}^{k} U_i^2 \sim \chi^2_{k-1}$. In our case with 3 columns, we would have $k = 3$, so that $\sum_{i=0}^{2} U_i \sim \chi^2_2$. Note that for independently and identically distributed Normal $(0, 1)$ random variables, say $V_i \sim$ Normal $(0, 1)$, that $\sum_{i=0}^{n} V_i^2 \sim \chi^2_n$; for our related U_j's, the sum of squares follows a χ^2 distribution with one fewer degrees of freedom than members.

In the Armitage paper, no proof is given of this fact, that is, no proof or derivation is given for $\sum_{i=0}^{k} U_i^2 \sim \chi^2_{k-1}$. Furthermore, several other papers take this fact as given and do not prove the result. Therefore, while clearly a well established result, it is difficult to conclude whether thee fact that $\sum_{i=0}^{2} U_i^2$ follows a χ^2_2 distribution is because of the relationships among the U_j's ascribed by the relatedness among the proportions of the columns and the nature of U_j, or for some other relatedness among the U_j's. If we assume that the above distribution of $\sum_{i=0}^{k} U_i^2$ is due to the relationships among the columns, then we can derive a statistic based on P_j to measure the probability of any result for alternative hypotheses, which we will do subsequently.

Consider the following setup for an alternative hypothesis. Let Table 2.3 give the proportions of the three genotypes for each of the cases and controls

	AA	Aa	aa
Case	p_0	p_1	p_2
Control	q_0	q_1	q_2

Table 2.3: Nomenclature for Alternative Hypothesis Probabilities in Contingency Table with One SNP.

That is, $p_j = P(X = j|Y = 1)$ and $q_j = P(X = j|Y = 0)$. With these probabilities, we can derive the expected proportion of cases for a given genotype as

follows

$$\begin{aligned} P(Y=1|X=j) &= \frac{P(Y=1)}{P(X=j)}P(X=j|Y=1) \\ &= \frac{\frac{n_{1,+}}{n}}{P(X=j|Y=1)P(Y=1)+P(X=j|Y=0)P(Y=0)}p_j \\ &= \frac{\frac{n_{1,+}}{n}}{p_j\frac{n_{1,+}}{n}+q_j\frac{n_{0,+}}{n}}p_j \\ &= \frac{n_{1,+}p_j}{n_{1,+}p_j+n_{0,+}q_j} \end{aligned}$$

Therefore, in a manner similar to equation (2.3.11), the distribution of $n_{1,j}$ is

$$f(n_{1,j}|n_{+,j},n_{i,+},n,p_j,q_j) = \frac{n_{+,j}!}{n_{1,j}!n_{0,j}!}\left(\frac{n_{1,+}p_j}{n_{1,+}p_j+n_{0,+}q_j}\right)^{n_{1,j}}\left(\frac{n_{0,+}q_j}{n_{1,+}p_j+n_{0,+}q_j}\right)^{n_{0,j}} \tag{2.3.14}$$

and therefore, calculating the expected value and variance in the usual way, we can calculate a new U_j as follows

$$U_j = \frac{\frac{n_{1,j}}{n_{+,j}} - \frac{n_{1,+}p_j}{n_{1,+}p_j+n_{0,+}q_j}}{\sqrt{\frac{1}{n_{1,+}}(\frac{n_{0,+}p_j}{n_{0,+}p_j+n_{0,+}q_j})(\frac{n_{0,+}q_j}{n_{1,+}p_j+n_{0,+}q_j})}} \tag{2.3.15}$$

Therefore, although we haven't proven it formally, we will consider that for the alternative frequencies given in Table 2.3, that $\sum_{i=0}^{2} U_i^2 \sim \chi_2^2$, with U_j definited as in (2.3.15). This will allow us to take either the probability of the test statistic, or the probability of the result, as being calculated from either the probability distribution function or the cumulative distribution function, respectively. Note that calculating the pdf of a 2 degree of freedom χ^2 statistic is easy, as it has pdf

$$\begin{aligned} f(x) &= \frac{1}{2^{\frac{k}{2}}\Gamma(\frac{k}{2})}x^{\frac{k}{2}-1}e^{\frac{-x}{2}} \\ &= \frac{1}{2}e^{\frac{-x}{2}} \end{aligned}$$

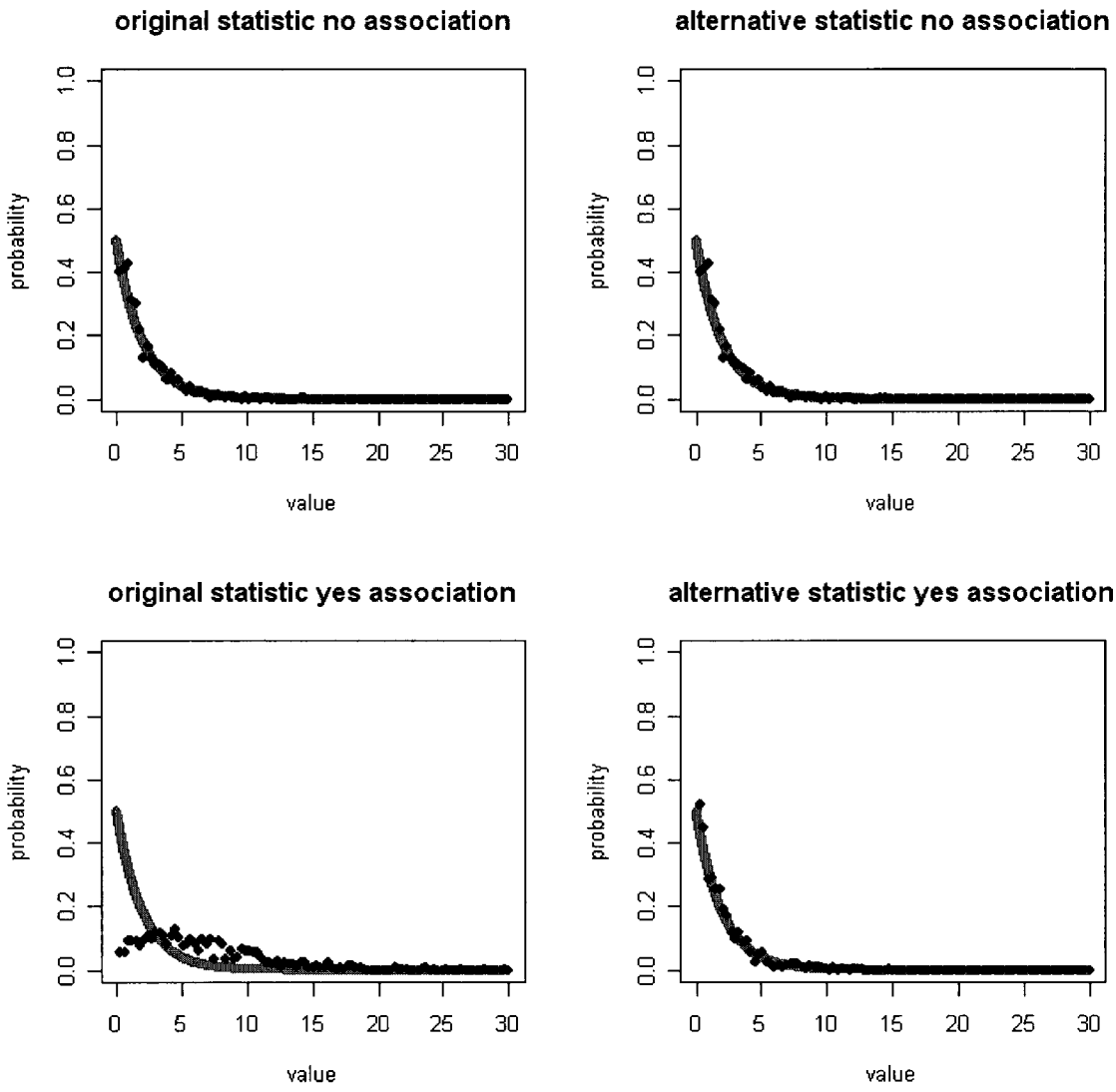

Figure 2.1: Estimates of Probability Density Functions for Statistics Used in Comparing Proportions. Columns give original statistic on the left (see Equation (2.3.13)) and new statistic on the right (see (2.3.15)). Rows give no association on the top ($RR1 = 1.0, RR2 = 1.0$) and association on the bottom ($RR1 = 1.25, RR2 = 1.5$), with both using a MAF of 0.25 and $P(D) = 0.10$ (for the use of these values in calculating the values in Table 2.3 see Section 2.3.5). Estimates of the probability density function were taken from 1000 random draws of 1000 individuals.

As a final motivating characterstic for the use of the above statistic, consider Figure 2.1, which compares the probability density function of a χ^2 distribution with 2 degrees of freedom to various empirical estimates of the probability distribution functions of the statistics given above. Note that the alternative statistic from (2.3.15), shown in the righthand column, matches up quite well under both an association and for no association, as expected.

Cochran-Armitage Test for Trend

There are at least two ways in which one can consider the generation of the trend statistic and p-value from the Cochran-Armitage Test for Trend. Although this may seem tedious, as with the Pearson's Chi Square Test it is necessary to understand the later work in the Methods chapter. The first version, described by Categorical Data Analysis by Agresti [2], is useful for understanding the formulae behind the trend line used in the test. The second version more closely follows the strategy of the Pearson subsubsection above, that of the Armitage 1955 paper [4], and includes derivations from a paper by Slager and Schaid [44].

The Cochran-Armitage Test for Trend works by looking for the presence of a linear trend among the probability of being a case given the genotype, across the genotypes. In other words, we have the probability of being a case given the genotype, $\pi_{1|j}$ for $j = 0, 1, 2$, which is equivalent to $\pi_{1|j} = P(Y = 1|X = j)$. We are looking for a linear equation, $\hat{\pi}_{1|j}$ of the form $\hat{\pi}_{1|j} = \alpha + \beta x_j$, which best approximates the observed $\pi_{1|j}$. Note that in this derivation, $\pi_{1|j}$ is equal to the observed $P(Y = 1|X = x)$, or in other words $\pi_{1|j} = \frac{n_{1,j}}{n_{+,j}}$; while $\hat{\pi}_{1|j}$ is the model fit. Note further that the x_j's represent 'scores', allowing us to set $x = [0, 1, 2]$ for the purposes of the trend test, so that $x_0 = 0, x_1 = 1, x_2 = 2$. As for the actual minimization, weighted ordinary least

squares regression is employed, where minimization is taken for the function

$$\begin{aligned} f(\alpha,\beta) &= \sum_{j=0}^{2} n_{+,j}\left(\hat{\pi}_{1|j} - \pi_{1|j}\right)^2 \\ &= \sum_{j=0}^{2} n_{+,j}\left(\alpha + \beta x_j - \frac{n_{1,j}}{n_{+,j}}\right)^2 \end{aligned} \tag{2.3.16}$$

with respect to α and β. To minimize the function, we take the partial derivatives of (2.3.16) with respect to α and β, which means we have to solve the system of linear equations

$$\begin{aligned} 0 &= \sum_{j=0}^{2} n_{+,j}\left(\alpha + \beta x_j - \frac{n_{1,j}}{n_{+,j}}\right) \\ 0 &= \sum_{j=0}^{2} n_{+,j}x_j\left(\alpha + \beta x_j - \frac{n_{1,j}}{n_{+,j}}\right) \end{aligned}$$

It is stated in Agresti 1990 [2] that the solution to the above system satisfies

$$\begin{aligned} \beta &= \frac{\sum_{j=0}^{2} n_{+,j}(\frac{n_{1,j}}{n_{+,j}} - \frac{n_{1,+}}{n})(x_j - \overline{x})}{\sum_{j=0}^{2} n_{+,j}(x_j - \overline{x})^2} \\ \alpha &= \frac{n_{1,+}}{n} - \beta\overline{x} \end{aligned}$$

where $\overline{x} = \frac{1}{n}\sum_{j=0}^{2} n_{+,j}x_j$. Agresti also states that the statistic

$$z^2 = \left(\frac{\beta^2}{(\frac{n_{1,+}}{n})(\frac{n_{0,+}}{n})}\right)\sum_{j=0}^{2} n_{+,j}(x_j - \overline{x})^2$$

satisfies a χ_1^2 distribution. Therefore, a p-value for the Cochran-Armitage Test for Trend, and hence the probability that the null hypothesis is true, is taken by comparing z^2 to a one degree of freedom Chi-Square distribution.

An alternative to the above can be considered along the lines of the method pre-

sented by Armitage [4] and used in the preceding subsubsection. Unlike the Pearson Chi-Square Test, in which formulae for alternative hypotheses were generated from scratch, here the rigorous derivations are available thanks to a paper by Slager [44]. If we work backwards from the statistic z presented above by subbing in β, we get the following

$$z = \sqrt{\left(\frac{\left(\frac{\sum_{j=0}^{2} n_{+,j}(\frac{n_{1,j}}{n_{+,j}} - \frac{n_{1,+}}{n})(x_j - \overline{x})}{\sum_{j=0}^{2} n_{+,j}(x_j - \overline{x})^2} \right)^2}{(\frac{n_{1,+}}{n})(\frac{n_{0,+}}{n})} \right) \sum_{j=0}^{2} n_{+,j}(x_j - \overline{x})^2}$$

$$z = \frac{\sum_{j=0}^{2} n_{+,j}(\frac{n_{1,j}}{n_{+,j}} - \frac{n_{1,+}}{n})(x_j - \overline{x})}{\sqrt{(\frac{n_{1,+}}{n})(\frac{n_{0,+}}{n}) \sum_{j=0}^{2} n_{+,j}(x_j - \overline{x})}}$$

$$z = \frac{\sum_{j=0}^{2} x_j(\frac{n_{0,+}}{n} n_{1,j} - \frac{n_{1,+}}{n} n_{0,j}) - \overline{x} \sum_{j=0}^{2} (\frac{n_{0,+}}{n} n_{1,j} - \frac{n_{1,+}}{n} n_{0,j})}{\sqrt{(\frac{n_{1,+}}{n})(\frac{n_{0,+}}{n}) \sum_{j=0}^{2} n_{+,j}(x_j - \overline{x})}}$$

$$z = \frac{\sum_{j=0}^{2} x_j(\frac{n_{0,+}}{n} n_{1,j} - \frac{n_{1,+}}{n} n_{0,j}) - 0}{\sqrt{(\frac{n_{1,+}}{n})(\frac{n_{0,+}}{n}) \sum_{j=0}^{2} n_{+,j}(x_j - \overline{x})}}$$

where we have $U = \sum_{j=0}^{2} x_j(\frac{n_{0,+}}{n} n_{1,j} - \frac{n_{1,+}}{n} n_{0,j})$ being the statistic, 0 being its expected value under the null hypothesis, and $\sqrt{(\frac{n_{1,+}}{n})(\frac{n_{0,+}}{n}) \sum_{j=0}^{2} n_{+,j}(x_j - \overline{x})}$ being the square root of the variance.

Under an alternative hypothesis where the probabilities of having a genotype based on being either case or control is as given in Table 2.3, then Slager [44] showed

that the expected value and variance of U are

$$\mathrm{E}(U) = n\left(\frac{n_{1,+}}{n}\right)\left(\frac{n_{0,+}}{n}\right)\sum_{j=0}^{2} x_j(p_j - q_j) \tag{2.3.17}$$

$$\begin{aligned}\mathrm{var}(U) =& n\left(\frac{n_{1,+}}{n}\right)\left(\frac{n_{0,+}}{n}\right)^2\left[\sum_{j=0}^{2} x_j^2 p_j - \left(\sum_{j=0}^{2} x_j p_j\right)^2\right] \\ &+ n\left(\frac{n_{1,+}}{n}\right)^2\left(\frac{n_{0,+}}{n}\right)\left[\sum_{j=0}^{2} x_j^2 q_j - \left(\sum_{j=0}^{2} x_j q_j\right)^2\right]\end{aligned} \tag{2.3.18}$$

By using the same trick as in the Pearson subsubsection, we can use the Central Limit Theorem to conclude that

$$\frac{U - \mathrm{E}(U)}{\sqrt{\mathrm{var}(U)}} \sim \mathrm{Normal}(0, 1)$$

and calculate the appropriate probabilities using either the pdf or cdf for various instances of alternative hypotheses.

2.3.5 Some Population Genetics

As was mentionned in the earlier subsection, Section 2.3.4, one can test for alternative hypotheses given expected genotype frequencies for case and control populations as given in Table 2.3. In this subsection, we briefly describe how to estimate those frequencies under certain assumptions. Note that the logic behind the following derivations is based on supplementary material provided with the R GeneticsDesign package [37].

The assumptions that we will need to estimate expected genotype frequencies are as follows. Let A be the minor allele, a the major allele, and let D be the disease. The first assumption we use is that in the population, the locus is in Hardy-Weinberg Equilibrium; that is $P(AA) = P(A)^2$, $P(Aa) = 2P(A)P(a)$ and $P(aa) = P(a)^2 =$

$(1 - P(A))^2$. As such, we can estimate the three genotype probabilities using either the minor or major allele frequency.

Next, we assume that the disease has a certain prevalence in the population, $P(D)$, and define the relative risks as being the following

$$RR_2 = \frac{P(D|AA)}{P(D|aa)}, RR_1 = \frac{P(D|Aa)}{P(D|aa)}$$

These relative risks are a common way of referencing the likelihood of developping disease based on genotype, and can be though of as to how many times more likely a homozygote major or heterozygote is to develop disease relative to the homozygote minor. Therefore

$$\begin{aligned} P(D) &= P(D|AA)P(AA) + P(D|Aa)P(Aa) + P(D|aa)P(aa) \\ &= RR_2P(D|aa)P(AA) + RR_1P(D|aa)P(Aa) + P(D|aa)P(aa) \\ \Rightarrow P(D|aa) &= \frac{P(D)}{RR_2P(AA) + RR_1P(Aa) + P(aa)} \end{aligned}$$

This allows us to calculate $P(D|Aa) = P(D|aa)RR_1$ and $P(D|AA) = P(D|aa)RR_2$, and furthermore, allows us to calculate the expected genotype frequencies for the disease group as follows

$$\begin{aligned} P(AA|D) &= \frac{P(AA)}{P(D)}P(D|AA) \\ &= \frac{RR_2P(AA)}{RR_2P(AA) + RR_1P(Aa) + P(aa)} \qquad (2.3.19) \\ P(Aa|D) &= \frac{RR_1P(Aa)}{RR_2P(AA) + RR_1P(Aa) + P(aa)} \qquad (2.3.20) \\ P(aa|D) &= \frac{P(aa)}{RR_2P(AA) + RR_1P(Aa) + P(aa)} \qquad (2.3.21) \end{aligned}$$

As for the probabilities of the genotypes given absence of the disease, then $P(AA|\overline{D}) =$

$\frac{P(AA)}{P(\overline{D})}P(\overline{D}|AA)$, and since $P(\overline{D}|AA) = 1 - P(D|AA)$, then

$$P(AA|\overline{D}) = \frac{P(AA)}{P(\overline{D})}(1 - P(D|AA))$$
$$= \frac{P(AA)}{1 - P(D)}\left(1 - \frac{RR_2P(D)}{RR_2P(AA) + RR_1P(Aa) + P(aa)}\right) \quad (2.3.22)$$
$$P(Aa|\overline{D}) = \frac{P(Aa)}{1 - P(D)}\left(1 - \frac{RR_1P(D)}{RR_2P(AA) + RR_1P(Aa) + P(aa)}\right) \quad (2.3.23)$$
$$P(aa|\overline{D}) = \frac{P(aa)}{1 - P(D)}\left(1 - \frac{P(D)}{RR_2P(AA) + RR_1P(Aa) + P(aa)}\right) \quad (2.3.24)$$

Therefore, for a given $P(D)$, $P(A)$, RR_1 and RR_2, we can estimate the expected genotype frequencies of Table 2.3, and hence calculate the distribution of the test statistic under alternative hypotheses.

2.3.6 Bayesian Inference

Although most of this Background chapter has been concerned with statistical inference arising from an interpretation of a p-value under a null hypothesis, in this thesis we will also be interested in Bayesian statistical inference. Unlike with frequentist statistical inference, Bayesian inference uses data to update a prior probability of association into a posterior probability of association. We will be particularly interested in this application with respect to hypothesis testing. Before we go on, consider the following quote, from Wacholder *et. al*, which they use as a preamble to their False Positive Report Probability [48].

> Classical frequentist statistical theory, which is most commonly taught in applied biostatistics courses, does not specifically address these probabilities. In classical theory, the truth of H_0 and H_A is considered unknown, not random. Therefore, we must go outside classical theory to consider H_0 and H_A probabilistically.

Note that our motivation for using Bayesian inference for hypothesis testing is the same as that which motivates the use of the False Positive Report Probability; however, we will follow a slightly different approach advocated by Lucke [30], as unlike the Wacholder paper, the Lucke paper is grounded in statistical theory.

Bayesian inference, as the name would suggest, is firmly rooted in Bayes Theorem, which states

$$P(A|B) = \frac{P(A)}{P(B)}P(B|A)$$

where A and B are events. Briefly, $P(B|A)$ is the probability of B given A, $P(B)$ is the probability of event B, $P(A)$ is the probability of event A, and $P(A|B)$ is the probability of A given B. In Bayesian statistical inference, we are typically concerned with determining whether or not the null hypothesis H_0 and an alternative hypothesis H_1 are more likely based on some observed statistic T. That is, assuming some prior belief of association, $P(H_0)$, and $P(H_1) = 1 - P(H_0)$, and by calculating the probability of seeing a statistic under the null and alternative hypothesis, $P(T|H_0)$ and $P(T|H_1)$, we can calculate the posterior probability as follows

$$P(H_0|T) = \frac{P(H_0)}{P(T)}P(T|H_0) = \frac{P(H_0)P(T|H_0)}{P(H_0)P(T|H_0) + P(H_1)P(T|H_1)} \tag{2.3.25}$$

Note that the notation used here is such that $P(H_0) = P(H_0\,\text{istrue})$ and $P(H_0|T) = P(H_0\text{true}|T = t)$ for some observed t.

If, alternatively, we do not have just one alternative hypothesis but multiple alternative hypotheses, we can use a slightly different methodology. Suppose that we assume that under the alternative hypothesis, some parameter θ, which can influence the distribution of the statistic T, is drawn from some prior distribution $P(\theta|H_1)$. Then the probability $P(T|H_1)$ can be estimated by $\int P(T|\theta)P(\theta|H_1)d\theta$, and so we

can update (2.3.25) as follows

$$P(H_0|T) = \frac{P(H_0)P(T|H_0)}{P(H_0)P(T|H_0) + P(H_1)\int P(T|\theta)P(\theta|H_1)d\theta} \tag{2.3.26}$$

We will use (2.3.26) later on in the Methods chapter as a method for assessing the probability a SNP is associated with disease given the genotyping results for that SNP.

2.4 Pattern Recognition and Machine Learning

Broadly speaking, pattern recognition is the task of identifying relationships and making predictions based on data. Machine learning is focused on the development of mathematical algorithms which 'learn' based on data. This section will focus on the subfield of machine learning known as supervised learning, which is concerned with approximating some sort of target function or classifier using a training dataset for which the correct output or classification is known. The purpose of this section is to introduce the relevant terminology and background to facilitate the later introduction of a classification algorithm in the Methods chapter.

This section will be organized as follows. The first subsection will introduce the relevant nomenclature of machine learning, and discuss idealized classifiers and their error. The second subsection will discuss the limit performance of classifiers, a topic dealt with by statistical consistency. Finally, the last subsection will deal with methods for evaluating the empirical performance of classifiers on given datasets.

2.4.1 Introduction

Let us consider more formally the data that will be used to make predictions. When we consider objects, and the search for shared characteristics, we are interested in the observations we can make of that object, which we consider as an m-dimensional

vector x, drawn from some space $\mathcal{X}$. Each dimension of x can represent either continuous, discrete or categorical data, making $\mathcal{X}$ some combination of data types. For any object we will also have a class label, $y \in \mathcal{Y}$, where $\mathcal{Y} = \{0, 1, 2, ..., C\}$, which may or may not be known at certain stages of analysis.

For the purpose of this thesis, and with an eye on the eventual dataset we are going to apply these methods on, we will now consider the following restrictions on $\mathcal{X}$ and $\mathcal{Y}$. First, we will assume that there are only two classes; that is, we have the binary classification problem $\mathcal{Y} = \{0, 1\}$. This is not a major restriction, as most of the work which follows can easily be extended to the general classification problem with C classes. Second, we will consider $\mathcal{X}$ to contain only discrete valued data. This is a somewhat more restrictive assumption, which simplifies the explanation of some of the probability notions which follow shortly, but for which the analysis of the limit behaviour will differ from the continuous case. Regardless, since we are eventually interested in analyzing discrete valued datasets, it is more useful to consider discrete valued data.

There are several ways in which a probability space could be defined for us to consider analysis, but we will use the following. On the domain $\mathcal{X} \times \mathcal{Y}$, we can consider an associated σ-algebra $\mathcal{A}$ of subsets of $\mathcal{X} \times \mathcal{Y}$. Since $\mathcal{X}$ is discrete, then we can consider $\mathcal{A}$ as the collection of all subsets of $\mathcal{X} \times \mathcal{Y}$. We can then define probability measures $\mu(x) = P(X = x)$ and $\eta(x) = P(Y = 1|X = x)$, so that

$$P(X = x, Y = y) = \begin{cases} \mu(x)\eta(x), & y = 1 \\ \mu(x)(1 - \eta(x)), & y = 0 \end{cases}$$

Using this setup, we can differentiate easily between the probability of an observation, $\mu(x)$, and the probability that for an observation it has class 1, $\eta(x)$. By differentiating between the two, we can focus on the more 'important' of the two, $\eta(x)$, which is more important in that it is more closely related to the task of assigning classes to

observations, which is our underlying goal. The triple $(\mathcal{X} \times \mathcal{Y}, \mathcal{A}, (\mu, \eta))$ therefore forms our probability space, and will be the space in which we conduct our analysis.

For further information on probability spaces, or any other topic from machine learning presented in this section, please consult A Probabilistic Theory of Pattern Recognition, by Devroye, Györfi and Lugosi [14].

Bayes Error

Consider a function $g : \mathcal{X} \to \mathcal{Y}$, which is a classifier assigning to any $x \in \mathcal{X}$ a value $y \in \mathcal{Y}$. Note that in this thesis we will usually refer to a classifier as begin a particular function, while a classification algorithm is a method or set of rules for constructing classifiers. For any particular classifier g, and for any pair $(x, y) \in \mathcal{X} \times \mathcal{Y}$, g is correct if $g(x) = y$ and incorrect if $g(x) \neq y$. For a fixed $\mathcal{X} \times \mathcal{Y}$ and a set of probability measures (μ, η), the probability of incorrect classification, or the generalization error for any classifier g is

$$L(g) = P(g(X) \neq Y)$$

This allows us to rank classifiers for a particular (μ, η) by saying that g is better than g' if $L(g) \leq L(g')$. If we consider the entire set of classifiers which are possible, $\mathcal{G} = \{g : \mathcal{X} \to \mathcal{Y}\}$, then the best possible classifier given (μ, η) is calculated as

$$g^* = \underset{g \in \mathcal{G}}{\operatorname{argmin}} L(g) \tag{2.4.1}$$

It turns out that the best function g^* is rather simple, and has the following form

$$g^*(x) = \begin{cases} 1, & \eta(x) > \frac{1}{2} \\ 0, & \eta(x) \leq \frac{1}{2} \end{cases} \tag{2.4.2}$$

Theorem 2.4.1 *g^* as defined in (2.4.2) satisfies (2.4.1).*

Proof: For any $g \in \mathcal{G}$,

$$
\begin{aligned}
& L(g^*) - L(g) \\
&= P(g^*(X) \neq Y) - P(g(X) \neq Y) \\
&= \sum_{(x,y)\in(\mathcal{X}\times\mathcal{Y})} (I_{\{g^*(x)\neq y\}} - I_{\{g(x)\neq y\}}) P(X = x, Y = y) \\
&= \sum_{x\in\mathcal{X}} \Big((I_{\{g^*(x)\neq 0\}} - I_{\{g(x)\neq 0\}})(1 - \eta(x)) + (I_{\{g^*(x)\neq 1\}} - I_{\{g(x)\neq 1\}})\eta(x) \Big) \mu(x) \\
&= \sum_{x\in\mathcal{X}} \left(I_{\{g^*(x)=1, g(x)=0\}} - I_{\{g^*(x)=0, g(x)=1\}} \right) (1 - \eta(x))\mu(x) \\
&+ \sum_{x\in\mathcal{X}} \left(I_{\{g^*(x)=0, g(x)=1\}} - I_{\{g^*(x)=1, g(x)=0\}} \right) \eta(x)\mu(x) \\
&= \sum_{x\in\mathcal{X}} \Big(I_{\{g^*(x)=1, g(x)=0\}}(1 - 2\eta(x)) + I_{\{g^*(x)=0, g(x)=1\}}(2\eta(x) - 1) \Big) \mu(x) \\
&\leq \sum_{x\in\mathcal{X}} \left(I_{\{g^*(x)=1, g(x)=0\}}(1 - 2(\frac{1}{2})) + I_{\{g^*(x)=0, g(x)=1\}}(2(\frac{1}{2}) - 1) \right) \mu(x) \\
&= 0
\end{aligned}
$$

since $\eta(x) > \frac{1}{2}$ when $g^*(x) = 1$ and $\eta(x) \leq \frac{1}{2}$ when $g^*(x) = 0$. Therefore, $L(g^*) - L(g) \leq 0 \Rightarrow L(g^*) \leq L(g)$, as required. ∎

The function g^* from above is called the Bayes Decision Rule or Bayes Classifier, and the associated error $L^* = L(g^*)$ is called the Bayes Error. In most situations, the probability measures (μ, η) are unknown, so the Bayes Classifier and Bayes Error are unknown. Trying to find classifiers which best approximate the Bayes Classifier over some known space $\mathcal{X} \times \mathcal{Y}$ and with unknown probability measures is dealt with by supervised learning.

Supervised Learning

Here, we are trying to build classifiers using data drawn from $\mathcal{X} \times \mathcal{Y}$ according to (μ, η). The training set S_n consists of n draws of a random variable, $(X_1, Y_1), \ldots (X_n, Y_n)$, which are assumed to be independently and identically distributed according to (μ, η). Note that we can speak of either a training set as the random variable S_n drawn with respect to (μ, η) on the space $(\mathcal{X} \times \mathcal{Y})^n$, or as a particular set of n values $s_n = \{(x_1, y_1), \ldots (x_n, y_n)\}$.

Individual functions which are built from a particular training set are denoted as $g_n = g_n(x; s_n) : \mathcal{X} \times (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{Y}$. For a particular function, we say that it's conditional probability of error is given by

$$L_n = L(g_n) = P(g_n(X; S_n) \neq Y | S_n)$$

Note that this is a random variable which depends on the training sample S_n, and as such, it is useful when we talk about a particular classifier. When we begin to evaluate methods to compare classification algorithms, $E(L_n)$ becomes useful as well, as it allows us to investigate the limit behaviour of the expectation of the classification algorithm, regardless of a specific draw. Investigating the limit behaviour of a classification algorithm is a topic dealt with by the subject of statistical consistency.

2.4.2 Statistical Consistency in Machine Learning

Consistency is a very desirable property in statistics. There are two forms of consistency; weak consistency or convergence in probability; and strong consistency, or almost sure convergence. With weak consistency, the *expected value* of a sequence of estimators will approximate a parameter of the underlying distribution with arbitrary precision as the sample size tends to infinity. With strong consistency, the *sequence of estimators* itself will approximate the parameter of the underlying distribution with

arbitrary precision as the sample size tends to infinity.

Returning to the setting at hand, consider classifiers constructed according to some classification algorithm with regard to a particular distribution (μ, η). Recall that for a particular classifier, we can consider its conditional probability of error as given by $L_n = P(g_n(X; S_n) \neq Y|S_n)$. Therefore, probabilities involving L_n must include the calculation of probabilities to be performed over the space of possible draws of S_n.

Definition 2.4.2 *A decision rule for constructing classifiers is weakly consistent for a particular distribution* (μ, η) *if for every* $\epsilon > 0$,

$$\lim_{n\to\infty} P(|L_n - L^*| > \epsilon) = 0$$

A decision rule is strongly consistent with respect to a particular distribution (μ, η) *if for every* $\epsilon > 0$,

$$P(\lim_{n\to\infty} |L_n - L^*| > \epsilon) = 0$$

A decision rule is universally weakly (strongly) consistent if it is weakly (strongly) consistent for all possible distributions (μ, η) *on the given sample space.*

2.4.3 Analyzing the Performance of a Classification Algorithm

Compared to analyzing the limit behaviour of a classification algorithm, analyzing the performance of a particular binary classifier is a relatively straightforward endeavour. Consider a classifier which has been used to predict classes for some test set, for which the real class membership is known. Then for all the samples, one of the conditions in Table 2.4 will hold.

The following quantities can then be generated using the nomenclature of Table

		Real	
		$Y = 1$	$Y = 0$
Predicted	$Y = 1$	True Positive (TP)	False Positive (FP)
	$Y = 0$	False Negative (FN)	True Negative (TN)

Table 2.4: Nomenclature for Binary Classification Test Outcomes.

2.4

$$\text{sensitivity} = \frac{TP}{TP + FN}$$
$$\text{specificity} = \frac{TN}{TN + FP}$$
$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Suppose further that the binary classification algorithm under consideration is set up so that it uses some sort of threshold in its classification. For algorithms with such a threshold a Receiver Operator Characteristic (ROC) curve can be constructed by varying the threshold for classification from $-\infty$ to $+\infty$, and plotting the resulting sensitivities and specificities achieved for each threshold value on a graph of sensitivity vs. $1 -$ specificity.

Cross-Validation

Whenever a classification algorithm is used, it is usually desirable to generate some sort of estimate as to its performance or generalization error. When a test set separate from the initial training set is available, then estimating performance can be done rather easily. However, if a separate test set is not available, it is usually better to obtain internal estimates of classifier performance using the whole training set and some sort of re-sampling procedure, as opposed to formally breaking the training set into separate training and testing components.

A common method to estimate training set performance is to use Cross-Validation

(CV). CV is relatively commonly used, and most texts which discuss either model building (ex: Regression Modeling Strategies by Harrell [19]) or machine learning (ex:Machine Learning by Mitchell [32]) will feature some sort of discussion on CV. Usually, one speaks of K-Fold CV, where the training set is split into K folds, each of which is of an approximately equal size. Each of these K folds is then used in turn as a test set for a classifier trained on the remaining $K-1$ folds of the training set. Each of the training set sample members can be tested in this way, allowing all training set members to be tested on a classification algorithm unbiased by it.

Depending on the computational requirements of the classification algorithm, K may be either small or large. In Leave-One-Out CV, K is set to equal n, the number of training set members, so that each training set sample is trained on a classifier using the other $n-1$ samples. Usually, this is only an option for classification algorithms which are not computationally expensive to run. In standard practice, K is normally set equal to 10, although 10 has no special significance. For a further discussion on the choice of K and its effects of estimates of error, and a validation of 10 as a common choice, please see Hastie *et al.* [20].

Chapter 3

Methods

The purpose of the Background chapter was to introduce the relevant genomic and statistical concepts which are necessary to understand the analysis of a Genome Wide Association (GWA) study. This chapter will focus on integrating those topics with other methods developed during the course of this thesis for the application of a classification algorithm to GWA data.

The chapter will be organized as follows. The first section will see the introduction of a filter which will act as a quality control check on the data. The filter will be used to select a subset of the genetic data for classification purposes with an aim at removing some of the redundancy between SNPs in the dataset. The second section will introduce the Naive Bayes classification algorithm, including a derivation of the classifier from first principles, as well as a modified version which weights the effect of a locus by the posterior probability of association. The third section will explain how the algorithm accomodates genetic data, and how the posterior probabilities are generated. Finally, the fourth and final section will prove the statistical consistency of the modified Naive Bayes algorithm under consideration.

3.1 r^2 Filter

Generally speaking, having a model with multiple variables which are highly correlated is not desirable. Aside from simply increasing the computational burden of analysis, it can lead to a loss of power in traditional statistical analyses, if some form of correction for multiple testing is applied with respect to the number of variables employed. In machine learning, it may cause the effect of a variable to be underranked, if the algorithm believes that the signal is coming from multiple inputs and not just one. Even worse, in a Naive Bayes estimator without feature selection, multiple highly correlated inputs may artificially inflate the effect of one locus and hamper classification accuracy.

Therefore, before considering the application of a classification algorithm to the dataset, we sought to remove the most highly correlated SNPs from further analysis. To do this, we decided to bin together SNPs using the r^2 measure. Recall that r^2, introduced in Section 2.3.3, is a measure of the linkage between two SNPs, scaled so that an r^2 of 0 indicates no linkage and an r^2 of 1 indicates perfect linkage, such that two SNPs are always inherited in tandem.

In the literature, r^2 filters have usually been used with respect to recommendations for DNA microarray design. For example, in Carlson *et al.* [11], they describe a greedy algorithm they devised to select a minimal set of SNPs which tag a larger set of SNPs at a prespecified r^2 threshold. This is relevant in the concept of DNA microarray design as it allows for the estimation of how many and which SNPs an array should contain to minimally tag a set of known population variants for a given r^2 threshold. Note that this approach is based on HapMap data [25], the online database which serves as an authoritative source on common genetic variants for several populations of interest, such as Africans (Yoruba from Nigeria) or Caucasians (people of North-Western European descent from Utah).

The Carlson algorithm works as follows. For all SNPs on a chromosome which

meet a prespecified minor allele frequency threshold, the SNP which exceeds the pairwise r^2 with the largest number of other SNPs is selected and binned with the SNPs it tagged. For all the SNPs in the bin, other SNPs are sought which also tag the bin by having a pairwise r^2 greater than the threshold with all other SNPs in the bin; these are the 'tag SNPs' for the bin. Next, one tag SNP was selected for each bin based on other criteria, such as considerations as to the genomic sequence for ease of assay hybridization, whether or not the SNP had a functional role (coding/exonic/genic/intergenic), or some other criteria. This bin of SNPs was then removed and the next most informative SNP sought, until the list of SNPs on the chromosome had been exhausted and a set of tag SNPs compiled. Mathematically, if we consider the set of all SNPs under consideration, M, then this algorithm creates a subset $M_{r^2_{threshold}} \subset M$ such that for any SNP $i \in M$ there exists a SNP $j \in M_{r^2_{threshold}}$ such that $r^2(i,j) \geq r^2_{threshold}$.

For our purposes, we sought to employ a slightly different algorithm since we were dealing with actual genome wide association data and not just theoretical assay design and HapMap samples. The algorithm we employed is described below in pseudocode as Algorithm 1. It is similar to the above except in two regards. First, for ease of computation, it scales across the chromosomes one end to the other, attempting to bin together SNPs only if they are within a prespecified window length. We set this window size at 1 Million Base Pairs (MBp), as it is very unlikely that SNPs at least 1MBp apart will recombine with an appreciable r^2, as over the course of the evolution of the population natural recombination events will decrease mutual inheritance of loci that far apart.

We also decided to change the binning requirement so that the pairwise r^2 between *all* SNPs in the bin had to exceed the r^2 threshold. The reason for this is twofold. Firstly, it allows for any SNP in a bin to tag the bin, so that when multiple studies are considered, if the SNP set to represent the bin in one study is not present in another study or does not meet quality control, then another SNP can

be taken from the bin to represent the locus. Consider for example coronary artery disease, whose most significantly associated locus is at 9p21.3. In this region, different highly correlated SNPs have been reported across studies as tagging the locus; rs10757274 (McPherson [31]), rs1333049 (the Wellcome Trust Case Control Consortium (WTCCC) [51])), rs4977574 (Myocardial Infarction Genetics Consortium [33]), rs10757278 (Helgadottir [21]). The algorithm we employ ensures that any of these SNPs could serve as the 9p21 tag SNP.

Secondly, since all SNPs in the bin tag every other SNP, it meant that any of the SNPs could be taken as the tag SNP. As such, instead of being worried about selecting a tag SNP which best represented the bin, we were free to select a tag SNP for which we had high confidence in genotyping. In this work, we selected as a tag SNP the SNP in the bin with the highest call rate among control genotype samples, which is related to the confidence that the genotype calling algorithm has in having assigned correct genotypes at that locus.

Once a subset of the data has been taken with the r^2 filter, classification algorithms can be used, which will be discussed subsequently in the section on the Naive Bayes classifier.

3.2 Naive Bayes Classifier

A Naive Bayes classifier is constructed around a simple premise; that the variables being considered in the model are independent. The aim of this section is to derive the classification algorithm mathematically, and to introduce a modification so that the effects of a locus are weighted by the posterior probability of association.

Before we begin, let us recall some of the nomenclature introduced in the machine learning section, Section 2.4. The setting involves a given training sample, $s_n = \{(x_1, y_1), \ldots (x_n, y_n)\}$, where each of the samples (x_i, y_i) comes from $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ is an m-dimensional space of discrete valued genotypes and $\mathcal{Y} = \{0, 1\}$, where 0

Algorithm 1 r^2 Filter

Require: control SNP data, $r^2_{threshold}$, SNP physical locations, windowSize

```
tagSNPset = empty
for chromosome = 1 to 22 do
    unselectedSNPs = all SNPs on chromosome
    while size(unselectedSNPs)> 0 do
        currentSNP = SNP from unselectedSNPs with lowest physical position
        currentBIN = currentSNP
        for all unselectedSNPs within windowSize of currentSNP do
            testSNP = next SNP from unselectedSNPs
            if r^2(binSNP,testSNP) ≥ r^2_threshold ∀ binSNP in currentBIN then
                currentBIN = currentBIN ∪ testSNP
            end if
        end for
        tagSNP = SNP with highest call rate in currentBin
        tagSNPset = tagSNPset ∪ tagSNP
    end while
end for
return tagSNPset
```

represents the class label of control and 1 represents the class label of CAD/case. Each of the samples (x_i, y_i) are sampled with respect to the underlying pair of probability measures (μ, η), such that $\mu(x) = P(X = x)$ and $\eta(x) = P(Y = 1|X = x)$.

3.2.1 Theoretical Foundations

The construction of the Naive Bayes classifier begins with the simple consideration of which is more likely, that the subject is a case or a control given the observed

genotype, or in other words, determining whether

$$\frac{P(Y = 1|X_i = x_i, 1 \leq i \leq m)}{P(Y = 0|X_i = x_i, 1 \leq i \leq m)} \gtrless 1 \tag{3.2.1}$$

Note that the symbol $\gtrless$ will be used to indicate that we are trying to determine which of the greater than or less than inequalities is true. To use this equation, we need to be able to evaluate the probability of having a class label of case or control based on the entire genotype. Practically speaking this is impossible, as forming accurate predictions of probabilities for 3^m genotypes when $m \geq 100,000$ would require an impossibly large training set. The 'naive' assumption in Naive Bayes allows us to rework the above probabilities in a manner which considers the probability of being a case or control based on genotypes at individual SNP loci, where we will make use of the naive assumption that

$$P(X_i = x_i|Y = y, X_j = x_j) = P(X_i = x_i|Y = y) \tag{3.2.2}$$

for any $i, j \in \{1, \ldots, m\}$, $y \in \{0, 1\}$. This assumption states that the probability of having a genotype at a particular SNP locus, given the class label of case or control, is independent of any other SNP locus; in other words, it assumes independence of the SNPs.

Now to rewrite our conditional probabilities from equation (3.2.1) using the above naive assumption. Consider first the probability of being a case based on a genotype. This is equivalent to

$$P(Y = 1|X_1 = x_1, \ldots, X_m = x_m) = \frac{P(Y = 1)P(X_1 = x_1, \ldots, X_m = x_m|Y = 1)}{P(X_1 = x_1, \ldots, X_m = x_m)} \tag{3.2.3}$$

since $P(A|B)P(B) = P(A \cap B) = P(B|A)P(A)$. This equation involves the conditional probability of having a genotype given a case label. We would like to use

the independence of the variables to rewrite this in terms of probabilities for individual SNPs. Consider $P(X_1 = x_1, \ldots, X_m = x_m | Y = 1)$. Given the conditional independence of the variables from equation (3.2.2), we can therefore write that

$$\begin{aligned}
&P(X_1 = x_1, \ldots, X_m = x_m | Y = 1) \\
&= P(X_1 = x_1 | Y = 1, X_2 = x_2, \ldots, X_m = x_m) P(X_2 = x_2, \ldots, X_m = x_m | Y = 1) \\
&= P(X_1 = x_1 | Y = 1) P(X_2 = x_2, \ldots, X_m = x_m | Y = 1) \\
&= \ldots \\
&= \prod_{i=1}^{m} P(X_i = x_i | Y = 1)
\end{aligned} \tag{3.2.4}$$

With the substitution from equation (3.2.4), equation (3.2.3) can be rewritten as

$$P(Y = 1 | X_1 = x_1, \ldots, X_m = x_m) = \frac{P(Y = 1) \prod_{i=1}^{m} P(X_i = x_i | Y = 1)}{P(X_1 = x_1, \ldots, X_m = x_m)} \tag{3.2.5}$$

Using the above equation (3.2.5), and the analogous result for controls, we can state that

$$\begin{aligned}
\frac{P(Y = 1 | X_1 = x_1, \ldots, X_m = x_m)}{P(Y = 0 | X_1 = x_1, \ldots, X_m = x_m)} &= \frac{\frac{P(Y=1) \prod_{i=1}^{m} P(X_i = x_i | Y=1)}{P(X_1 = x_1, \ldots, X_m = x_m)}}{\frac{P(Y=0) \prod_{i=1}^{m} P(X_i = x_i | Y=0)}{P(X_1 = x_1, \ldots, X_m = x_m)}} \\
&= \frac{P(Y = 1) \prod_{i=1}^{m} P(X_i = x_i | Y = 1)}{P(Y = 0) \prod_{i=1}^{m} P(X_i = x_i | Y = 0)}
\end{aligned} \tag{3.2.6}$$

As a result, the original equation of interest (3.2.1) can be rewritten as

$$\frac{P(Y = 1) \prod_{i=1}^{m} P(X_i = x_i | Y = 1)}{P(Y = 0) \prod_{i=1}^{m} P(X_i = x_i | Y = 0)} \gtrless 1 \tag{3.2.7}$$

The above equation (3.2.7) is the traditional form of the Naive Bayes classifier, which uses the probability of having a genotype at a locus conditional on the class label of case or control. However, it is more useful for us to work with the alter-

native conditional probability; the probability of having a class identity of case or control conditional to the genotype. This will allow us to choose between alternative methods for determining the probability of class identity of case or control based on genotype. For example, an additive model assumes that there is a linear trend in the probability of being a case based on genotype across the genotypes of homozygote major, heterozygote and homozygote minor, while a genotype model allows each of the three genotypes to have their own risks independent of the others. These models will be discussed in more detail in subsequent subsections.

Now, in rewriting equation (3.2.7) , consider the following

$$P(X_i = x_i|Y = 1) = \frac{P(X_i = x_i)}{P(Y = 1)} P(Y = 1|X_i = x_i) \tag{3.2.8}$$

since $P(A|B)P(B) = P(A \cap B) = P(B|A)P(A)$. Then we can rewrite equation (3.2.7) using the above equation and the analogous equation for controls to give

$$\begin{aligned} 1 &\gtrless \frac{P(Y = 1) \prod_{i=1}^{m} \frac{P(X_i=x_i)}{P(Y=1)} P(Y = 1|X_i = x_i)}{P(Y = 0) \prod_{i=1}^{m} \frac{P(X_i=x_i)}{P(Y=0)} P(Y = 0|X_i = x_i)} \\ &= \left(\frac{P(Y = 0)}{P(Y = 1)}\right)^{m-1} \left(\prod_{i=1}^{m} \frac{P(Y = 1|X_i = x_i)}{P(Y = 0|X_i = x_i)}\right) \end{aligned} \tag{3.2.9}$$

which is the Naive Bayes classifier using the probability of being a case or control conditional to the genotype.

Equation (3.2.9) gives a traditional form of the Naive Bayes classifier. Now consider the following modification, which attempts to weight the probability for each locus depending on the probability that the result seen at that locus is real. Much like how rewriting Naive Bayes to use the probability of being a case conditional on genotype allows for the consideration of different methods of modeling the probability, the following will allow for the weighting of each locus according to both how likely the observed results are and a desired level of stringency.

Consider that for any locus, the locus either is or is not associated with disease; that is, either the null hypothesis of no association is true H_0^i, or false $\overline{H_0}^i = H_1^i$, with probability 1. However, since we do not know whether or not H_0^i or H_1^i is true, we can attempt to estimate the probability that either of them are true, using the ideas of Section 2.3.6. This will allow us to estimate $P(H_0^i)$ for each locus according to the principles of Bayesian inference (exactly how this is calculated will be described in Section 3.3). Note that notation-wise, while $P(H_0)$ refered to the prior probability of association in Section 2.3.6, in this chapter we will use $P(H_0^i)$ to refer to the posterior probability of association given statistic T_i, which we will estimate by $\hat{P}(H_0^i) = P(H_0^i|T_i = t)$. Before we consider estimating $P(H_0^i)$ in Section 3.3.1, remember that it is the probability that the locus is truly associated with disease.

Now, consider the probability $P(Y = 1|X_i = x_i)$ under two different scenarios; when the locus either is or is not genuinely associated with disease. That is, considering the probabilities of the null or alternative hypotheses given the data, and using the fact that $P(A) = P(A \cap B) + P(A \cap \overline{B}) = P(A|B)P(B) + P(A|\overline{B})P(\overline{B})$, we can therefore estimate $P(Y = 1|X_i = x_i)$ as

$$\begin{aligned} P(Y = 1|X_i = x_i) &= P(Y = 1|X_i = x_i, H_0^i)P(H_0^i) \\ &\quad + P(Y = 1|X_i = x_i, H_1^i)P(H_1^i) \end{aligned} \tag{3.2.10}$$

Now, since if H_0^i, then there is no association at the locus, then $P(Y = 1|X_i = x_i, H_0^i) = P(Y = 1)$, and so equation (3.2.10) can be rewritten as

$$\begin{aligned} P(Y = 1|X_i = x_i) &= P(Y = 1)P(H_0^i) \\ &\quad + P(Y = 1|X_i = x_i, H_1^i)P(H_1^i) \end{aligned} \tag{3.2.11}$$

Therefore, we can rewrite equation (3.2.9) as

$$1 \gtrless \left(\frac{P(Y=0)}{P(Y=1)}\right)^{m-1} \left(\prod_{i=1}^{m} \frac{P(Y=1)P(H_0^i) + P(Y=1|X_i=x_i, H_1^i)P(H_1^i)}{P(Y=0)P(H_0^i) + P(Y=0|X_i=x_i, H_1^i)P(H_1^i)}\right) \tag{3.2.12}$$

It is convenient computationally to rewrite the above (3.2.12) by taking its logarithm, which yields the equivalent equation

$$\begin{aligned} 0 \gtrless\ & (m-1)\log\left(\frac{P(Y=0)}{P(Y=1)}\right) \\ & + \sum_{i=1}^{m} \log\left(\frac{P(Y=1)P(H_0^i) + P(Y=1|X_i=x_i, H_1^i)P(H_1^i)}{P(Y=0)P(H_0^i) + P(Y=0|X_i=x_i, H_1^i)P(H_1^i)}\right) \end{aligned} \tag{3.2.13}$$

Now, consider the argument in the interior of the second logarithm of equation (3.2.13). Since $P(H_0^i) = 1 - P(H_1^i)$, and since $P(Y=1|X_i=x_i, H_1^i) = 1 - P(Y=0|X_i=x_i, H_1^i)$, we can rewrite the interior of the logarithm as follows

$$\begin{aligned} & \frac{P(Y=1)P(H_0^i) + P(Y=1|X_i=x_i, H_1^i)P(H_1^i)}{P(Y=0)P(H_0^i) + P(Y=0|X_i=x_i, H_1^i)P(H_1^i)} \\ &= \frac{P(Y=1)P(H_0^i) + P(Y=1|X_i=x_i, H_1^i)P(H_1^i)}{(1-P(Y=1))P(H_0^i) + (1-P(Y=1|X_i=x_i, H_1^i))P(H_1^i)} \\ &= \frac{P(Y=1)P(H_0^i) + P(Y=1|X_i=x_i, H_1^i)P(H_1^i)}{P(H_0^i) - P(Y=1)P(H_0^i) + P(H_1^i) - P(Y=1|X_i=x_i, H_1^i)P(H_1^i)} \\ &= \frac{1-(1-P(Y=1)P(H_0^i) - P(Y=1|X_i=x_i, H_1^i)P(H_1^i))}{1-P(Y=1)P(H_0^i) - P(Y=1|X_i=x_i, H_1^i)P(H_1^i)} \\ &= \frac{1}{1-P(Y=1)P(H_0^i) - P(Y=1|X_i=x_i, H_1^i)P(H_1^i)} - 1 \end{aligned}$$

and therefore, equation (3.2.13) can be rewritten as

$$\begin{aligned} 0 \gtrless\ & (m-1)\log\left(\frac{P(Y=0)}{P(Y=1)}\right) \\ & + \sum_{i=1}^{m} \log\left(\frac{1}{1 - P(Y=1)P(H_0^i) - P(Y=1|X_i = x_i, H_1^i)P(H_1^i)} - 1\right) \\ & = nbs(x) \end{aligned} \tag{3.2.14}$$

where we define $nbs(x)$ to equal the right side of the inequality in (3.2.14). Here, the function $nbs(x)$ stands for 'Naive Bayes Score', such that the value on the right side of (3.2.14) is termed the Naive Bayes Score for a person with genotype x. We can then build the Naive Bayes classifier g as a function which calls case when $nbs(x) \geq 0$ and control when $nbs(x) < 0$, as follows

$$g(x) = \begin{cases} 1, & nbs(x) \geq 0 \\ 0, & nbs(x) < 0 \end{cases} \tag{3.2.15}$$

Note that the Naive Bayes classifier is very similar in form to the Bayes Decision Rule from Section 2.4. Specifically, if $nbs(x)$ is a good approximation of $\eta(x)$, the two functions will be similar; in the case that there is perfect overlap, when $\{x|\eta(x) > \frac{1}{2}\} = \{x|nbs(x) > 0\}$, then the Naive Bayes classifier (3.2.15) *equals* the Bayes Decision Rule, and is the best classification algorithm possible. This is an important fact will be discussed more specifically in Section 3.4 on the statistical consistency of Naive Bayes classifiers.

One last refinement to the above classifier (3.2.15) can be made if one desires the minimization of certain types of classification errors. For example, if the misdiagnosis of a control as a case is more problematic than diagnosing a case as a control, then one would want to weight the classifier g to call cases at an $nbs(x)$ value less than 0.

Therefore, consider a further refinement to the Naive Bayes classifier where

$$g(x;\theta) = \begin{cases} 1, & nbs(x) \geq \theta \\ 0, & nbs(x) < \theta \end{cases} \tag{3.2.16}$$

The above formula (3.2.16) is the final form of the modified Naive Bayes classifier which will be used in classification. If $\theta = 0$, then the classifiers from (3.2.16) and (3.2.15) are the same.

To be able to use the above classifier, the training set has to be used to estimate $P(Y = 1)$ along with $P(H_0^i)$ and $P(Y = 1|X_i = x_i, H_1^i)$ for each of the loci $1 \leq i \leq m$. Generating $P(H_0^i)$ and $P(Y = 1|X_i = x_i, H_1^i)$ can be accomplished in several ways, depending on assumptions made and underlying inheritance patterns considered. The different methods of constructing these probabilities, and hence the Naive Bayes classifier, will be discussed subsequently.

3.3 Estimating Conditional Probabilities

Using the Naive Bayes classifier requires estimations of $P(Y = 1)$, along with $P(H_0^i)$ and $P(Y = 1|X_i = x_i, H_1^i)$ for each of the loci i. Estimating $P(Y = 1)$ is straightforward; in the simplest form, it can be estimated as

$$\hat{P}(Y = 1) = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3.3.1}$$

Alternatively, one could use the following,

$$\hat{P}(Y = 1) = \left(\prod_{i=1}^{m} \frac{n_{1,+}^i}{n^i}\right)^{\frac{1}{m}} \tag{3.3.2}$$

which is the geometric mean of the probability of being a case across each of the i SNPs. Note that these two values will be roughly identical, but the latter (3.3.2) is better able to accommodate genotyping errors when systematic errors in call rate between the case and control populations exist.

The estimation of $P(H_0^i)$ and $P(Y = 1|X_i = x_i, H_1^i)$ are more complicated and are done with respect to two underlying inheritance models. The first model, which shall be called the Genotype model of inheritance, works under the null hypothesis that $P(X_i = j|Y = 0) = P(X_i = j|Y = 1)$ for $j \in \{0, 1, 2\}$. Under the alternate hypothesis of association, these probabilities are not equal, so the related conditional probabilities $P(Y = 1|X_i = 0), P(Y = 1|X_i = 1)$ and $P(Y = 1|X_i = 2)$ are allowed to take different values. The second model, which shall be known as the Additive model of inheritance, assumes a linear trend in the proportion of being a case given a genotype across the three genotypes.

3.3.1 Posterior Probabilities of Association

For both the Genotype and Additive models of inheritance, estimates are needed for $P(H_0^i)$ for every SNP i. As mentionned earlier, we will do this using the principles of Bayesian Inference first presented in Section 2.3.6.

Consider Equation (2.3.26) rewritten for a particular SNP i; this would give that

$$P(H_0^i|T^i) = \frac{P(H_0)P(T^i|H_0)}{P(H_0)P(T^i|H_0) + P(H_1)\int P(T^i|\theta)P(\theta|H_1)d\theta} \tag{3.3.3}$$

Therefore, we will form the estimate $\hat{P}(H_0^i)$ from $P(H_0^i|T_i)$, the posterior probability of association. To evaluate Equation (3.3.3), we will need the distribution of the statistic under the null hypothesis, the distribution of the statistic under the alternative hypothesis, an assumed prior distribution on θ, and a prior probability of association for the null hypothesis. Specific values used as prior probabilities of

association will be given in the Application chapter. As to the distribution of the statitiscs under the null and alternative hypotheses, this will be discussed more in Section 3.3.2 and Section 3.3.4, while the priors for the two models for the parameter θ will be discussed in Section 3.3.3 and Section 3.3.5.

3.3.2 Genotype Model

As was stated before, the Genotype inheritance probability model allows for the three probabilities of being a case based on genotype, $P(Y = 1|X_i = 0), P(Y = 1|X_i = 1)$ and $P(Y = 1|X_i = 2)$ to be estimated separately under the alternate hypothesis H_1^i. This is done purposefully with the Pearson Chi-Square Test in mind, and the calculations from Section 2.3.4.

Consider the following nomenclature for a contingency table with one SNP, which is the same as that introduced in Section 2.3.4 which dealt with single variable tests of association.

	AA	Aa	aa	
Control	$n^i_{0,0}$	$n^i_{0,1}$	$n^i_{0,2}$	$n^i_{0,+}$
Case	$n^i_{1,0}$	$n^i_{1,1}$	$n^i_{1,2}$	$n^i_{1,+}$
	$n^i_{+,0}$	$n^i_{+,1}$	$n^i_{+,2}$	n^i

Table 3.1: Nomenclature for Contingency Table for SNP i. Similar to Table 2.2. Note that when referring to the results at a particular locus i, a superscript i is used.

We can calculate the probabilities under an assumption of association as

$$\hat{P}(Y = 1|X_i = j, H_1^i) = \frac{n^i_{1,j}}{n^i_{+,j}}$$

where we are motivated by the fact that a sample proportion converges to its underlying true population proportion as the sample size becomes arbitrarily large.

As for the statistic under consideration, recall from Equation (2.3.15), rewritten

for SNP i

$$U_j^i = \frac{\frac{n_{1,j}^i}{n_{+,j}^i} - \frac{n_{1,+}^i p_j^i}{n_{1,+}^i p_j^i + n_{0,+}^i q_j^i}}{\sqrt{\frac{1}{n_{1,+}^i}\left(\frac{n_{0,+}^i p_j^i}{n_{0,+}^i p_j^i + n_{0,+}^i q_j^i}\right)\left(\frac{n_{0,+}^i q_j^i}{n_{1,+}^i p_j^i + n_{0,+}^i q_j^i}\right)}} \tag{3.3.4}$$

which was shown to follow

$$\sum_{i=0}^{2} (U_j^i)^2 \sim \chi_2^2$$

From the population genetics laid out in Section 2.3.5, we can calculate $\{p_0^i, p_1^i, p_2^i\}$ and $\{q_0^i, q_1^i, q_2^i\}$ under the alternative and null hypotheses using: the major allele frequency (estimated from the control samples); the prior distribution of the relative risks (as given below in Section 3.3.3); and the probability of disease (taken as $P(D) = 0.1$). Taken together, this allows $\hat{P}(H_0^i)$ to be calculated for each SNP under the Genotype model.

Therefore, the Naive Bayes Score function for the genotype version of the modified Naive Bayes Classifier, using the geometric mean for $\hat{P}(Y = 1)$ is

$$\begin{aligned} nbs_{genotype}(x) =& (m-1)\log\left(\frac{\hat{P}(Y=0)}{\hat{P}(Y=1)}\right) \\ &+ \sum_{i=1}^{m} \log\left(\frac{1}{1 - \hat{P}(Y=1)\hat{P}(H_0^i) - \frac{n_{1,j}^i}{n_{+,j}^i}\hat{P}(H_1^i)} - 1\right) \end{aligned} \tag{3.3.5}$$

where $x_i = j$.

3.3.3 Genotype Model Prior

In the previous section, calculating $\hat{P}(H_0^i)$ was said to depend on the prior probability of the relative risks RR_1 and RR_2. For the purposes of this thesis, we chose to use a prior on the relative risks as illustrated in Figure 3.1. This prior was derived from multiplying together independent and identical priors which are shown in the single dimensional form in Figure 3.2. Please see Section 3.3.5 for a further computational

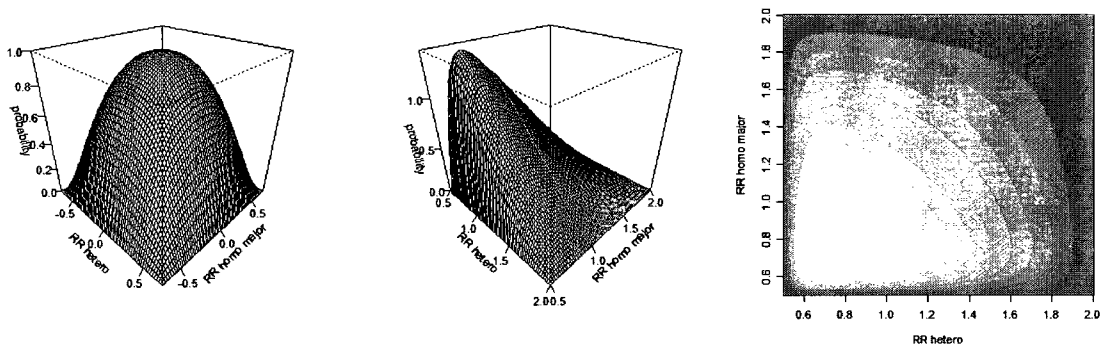

Figure 3.1: Prior Distribution of Relative Risks Under the Genotype Model. The left plot shows the probability distribution function for the logarithm of the relative risks. The middle plot shows the probability distribution function to the linear relative risks. The right plot shows a contout map of the middle plot.

and motivational background to the prior.

3.3.4 Additive Model

The Additive inheritance probability model is largely based around an additive model of inheritance, and the testing performed in the Cochran-Armitage Test for Trend, which was outlined in Section 2.3.4.

Recall that the Cochran-Armitage Test for Trend works by fitting a weighted ordinary least squares linear regression line through the proportions $P(Y = 1|X_i = j)$ for $j \in \{0, 1, 2\}$, which are estimated as $\frac{n^i_{1,j}}{n^i_{+,j}}$. It then looks for an α^i and β^i such that

$$\hat{P}(Y = 1|X_i = j, H^i_1) = \alpha^i + \beta^i j \tag{3.3.6}$$

It was further shown in Section 2.3.4 that the α^i and β^i which best fit the model are

given by

$$\beta^i = \frac{\sum_{j=0}^{2} n^i_{+,j}(\frac{n^i_{1,j}}{n^i_{+,j}} - \frac{n^i_{+,1}}{n^i})(j - \overline{x})}{\sum_{j=0}^{2} n^i_{+,j}(j - \overline{x})^2} \tag{3.3.7}$$

$$\alpha^i = \frac{n^i_{+,1}}{n^i} - \beta\overline{x} \tag{3.3.8}$$

where $\overline{x} = \frac{1}{n^i}\sum_{j=0}^{2}(n^i_{j,+})j$. Therefore, under an assumption of association, we can use the estimate of $\hat{P}(Y = 1|X_i = j, H_1^i)$ given in (3.3.6).

As for $\hat{P}(H_0^i)$, recall that for the statistic U from Section 2.3.4, rewritten for SNP i, that $U^i = \sum_{j=0}^{2} j(\frac{n^i_{0,+}}{n^i}n^i_{1,j} - \frac{n^i_{1,+}}{n^i}n^i_{0,j})$ and has expected value and variance given by

$$\begin{aligned}
\mathrm{E}(U^i) =& n\left(\frac{n^i_{1,+}}{n^i}\right)\left(\frac{n^i_{0,+}}{n^i}\right)\sum_{j=0}^{2} j(p^i_j - q^i_j) \\
\mathrm{var}(U^i) =& n\left(\frac{n^i_{1,+}}{n^i}\right)\left(\frac{n^i_{0,+}}{n^i}\right)^2\left[\sum_{j=0}^{2} j^2 p^i_j - \left(\sum_{j=0}^{2} j p^i_j\right)^2\right] \\
&+ n\left(\frac{n^i_{1,+}}{n^i}\right)^2\left(\frac{n^i_{0,+}}{n^i}\right)\left[\sum_{j=0}^{2} j^2 q^i_j - \left(\sum_{j=0}^{2} j q^i_j\right)^2\right]
\end{aligned}$$

so the statistic $T^i = \frac{U^i - \mathrm{E}(U^i)}{\sqrt{\mathrm{var}(U^i)}} \sim \mathrm{Normal}(0,1)$ can be used with $\{p^i_0, p^i_1, p^i_2\}$ and $\{q^i_0, q^i_1, q^i_2\}$ generated in the same fashion as in Section 3.3.2.

Therefore, using $\hat{P}(Y = 1|X_i = x_i)$ and $\hat{P}(H_0^i)$, the Naive Bayes Score function for the modified Naive Bayes classifier using the Additive model is

$$\begin{aligned}
nbs_{additive}(x) =& (m-1)\log\left(\frac{\hat{P}(Y=0)}{\hat{P}(Y=1)}\right) \\
&+ \sum_{i=1}^{m}\log\left(\frac{1}{1 - \hat{P}(Y=1)\hat{P}(H_0^i) - (\alpha^i + \beta^i j)\hat{P}(H_1^i)} - 1\right)
\end{aligned} \tag{3.3.9}$$

where $x_i = j$.

3.3.5 Additive Model Prior

The prior for the relative risk used in the Additive model is as given in Figure 3.2. It is the same as the single dimensional priors for the relative risks in the Genotype model given in Section 3.3.3. Note that the prior describes RR_2; for the additive model, RR_1 is set as being equal to $\frac{RR_2-1}{2}+1$.

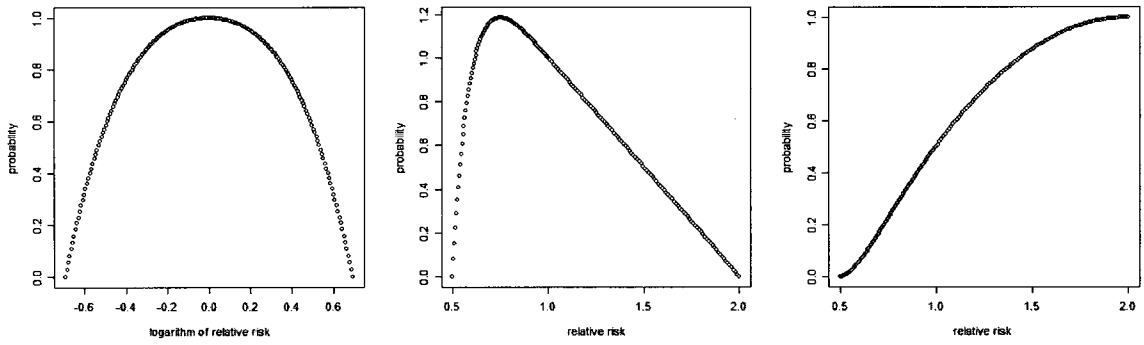

Figure 3.2: Prior Distribution of Relative Risk Under the Additive Model. The left plot shows the probability distribution function for the logarithm of the relative risk. The middle plot shows the probability distribution function for the relative risk on a linear scale. The right plot shows the cumulative distribution function for the relative risk on a linear scale.

Although the computation is quite simple in practice when trying to approximate the integral, in its original form it is quite complicated; namely, it has the probability distribution function

$$f_{RR}(rr) = \begin{cases} (2-\frac{1}{rr})\frac{1}{rr^2}, & \frac{1}{2} \leq rr < 1 \\ 2-rr, & 1 \leq rr \leq 2 \end{cases} \tag{3.3.10}$$

and the cumulative distribution function

$$F_{RR}(rr) = \begin{cases} \frac{-2}{rr} + \frac{1}{2rr^2} + 2, & \frac{1}{2} \leq rr < 1 \\ 2rr - \frac{1}{2}rr^2 - 1, & 1 \leq rr \leq 2 \end{cases} \tag{3.3.11}$$

Note that distribution of the logarithm is more pleasant, where if we take $y = \log(rr)$, we get that y follows a probability distribution function as follows

$$f_Y(y) = \begin{cases} (2 - e^y)e^y, & -\log(2) \leq y < 0 \\ (2 - e^{-y})e^{-y}, & 0 \leq y \leq \log(2) \end{cases} \tag{3.3.12}$$

Despite the seemingly odd nature of the prior, it was conceived with the following goals in mind: to be centered around a relative risk of 1, *i.e.* to have $F_{RR}(1) = \frac{1}{2}$; to give equal weights to relative risks above and below 1, *i.e.* so that for $rr \leq 1$, $F_{RR}(rr) = 1 - F_{RR}(\frac{1}{rr})$; to have a reasonable and finite upper and lower bound; and to be computationally amenable to integral approximations. This last point is due to the fact that the integral $\int P(T^i|RR)P(RR|H_1)dRR$ was thought to be too difficult to evaluate in a closed form solution due to the complex relationship among the probabilities p_j and q_j, relative risks and the distribution of the statistic for any choice of prior on the relative risks. Therefore, a relatively simple prior was sought which met the aforementioned goals, and under the original assumption that for $1 \leq rr \leq 2$, $f_{RR}(rr) = 2 - rr$. This sort of prior, in which not a lot of confidence was ascribed to the distribution under an alternative hypothesis, can best be described as a vague or non-informative prior.

3.4 Statistical Consistency of Naive Bayes Classifier

In Section 2.4, during the discussion of Machine Learning, the principle of statistical consistency was introduced with respect to classifiers. Here, it will be shown that the modified Naive Bayes classifier is consistent for a certain class of probability measures $(\mu(x), \eta(x))$.

During the analysis of consistency, we will require the use of some theorems

which deal with limit behaviour: particularly, we will need the Strong Law of Large Numbers, which is shown below.

Theorem 3.4.1 Strong Law of Large Numbers [12]. *Let $X_1, X_2, \ldots$ be iid random variables with $E(X_i) = \mu$ and $Var(X_i) = \sigma^2 < \infty$, and define $\overline{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$. Then, for every $\epsilon > 0$,*

$$P\left(\lim_{n\to\infty} |\overline{X}_n - \mu| < \epsilon\right) = 1$$

Proof: For a proof of the above, please see Casella and Berger [12]. ∎

Now we can investigate the Naive Bayes classifier. Consider the Genotype model.

Theorem 3.4.2 *The genotype modified Naive Bayes classification algorithm given in* (3.3.5) *is strongly consistent under the restriction on $\eta(x)$ given by* (3.2.2).

Proof:

Recall that for a particular pair of probability measures $(\mu(x), \eta(x))$, where $\mu(x) = P(X = x)$, $\eta(x) = P(Y = 1|X = x)$, that the restriction given by (3.2.2) states that

$$P(X_i = x_i|Y = y, X_j = x_j) = P(X_i = x_i|Y = y)$$

This restriction enabled us to assume independence between loci. To prove the above, all that is needed is to prove that in the limit case $\{x|\eta(x) \geq \frac{1}{2}\} = \{x|nbs_{genotype}(x) \geq 0\}$, as the Bayes Decision Rule g^* is the best possible classifier on $\mathcal{X} \times \mathcal{Y}$ and so $L^* = P(g^*(x) \neq Y)$.

Let $I = \{i|H_0^i \text{ is true}\}$, and let $J = \{i|H_1^i \text{is true}\}$. Note that $I \cup J = \{1, 2, \ldots, m\}$.

Consider the genotype Naive Bayes Score equation from Equation (3.3.5)

$$nbs_{genotype}(x) = (m-1)\log\left(\frac{\hat{P}(Y=0)}{\hat{P}(Y=1)}\right) + \sum_{i=1}^{m}\log\left(\frac{1}{1-\hat{P}(Y=1)\hat{P}(H_0^i) - \frac{n_{1,j}^i}{n_{+,j}^i}\hat{P}(H_1^i)} - 1\right)$$

Earlier, while discussing the construction of the genotype modified Naive Bayes classifier, it was mentionned that

$$\hat{P}(Y=1|X_i=j,H_1^i) = \frac{n_{1,j}^i}{n_{+,j}^i}$$

was chosen as in the limit case, it will converge to the true underlying proportion. Let us investigate this more clearly.

For a locus i, genotype j, and sample size n, let $K_{i,j} = \{k|x_i^k = j, 1 \leq k \leq n\}$; that is, $K_{i,j}$ is the subset of indicators of the samples from the training set who have genotype j at locus i. Consider the size of $K_{i,j}$ as $n \to \infty$. If $P(X = x|x_i = j) = 0$, then $|K_{i,j}| \equiv 0\ \forall n$. However, as the loss function for any classifier g, $L(g) = P(g(X) \neq Y)$ gives no weight to samples x for which $\mu(x) = 0$, then the loss function is only affected by samples for which $P(X = x|x_i = j) > 0$, and hence we need only be concerned with the loci i whose genotypes j allow $|K_{i,j}| \to \infty$ as $n \to \infty$.

Now, consider

$$\hat{P}(Y=1|X_i=j,H_1^i) = \frac{n_{1,j}^i}{n_{+,j}^i} = \frac{1}{n_{+,j}^i}\sum_{k\in K_{i,j}} y^k = \frac{1}{|K_{i,j}|}\sum_{k\in K_{i,j}} y^k = \overline{Y}_{|K_{i,j}|}$$

which is the mean of the class labels for samples with genotype j at locus i. As $|K_{i,j}| \to \infty$ as $n \to \infty$, then we can consider the class labels of samples with genotype j at locus i to be a series of iid random variables taken from a Bernoulli distribution with respect to the real underlying probability $p = P(Y = 1|X_i = j)$. Since the

expected values and variance of this distribution are p and $p(1-p)$ respectively, both finite, then by the Strong Law of Large Numbers, as $n \to \infty$, $|K_{i,j}| \to \infty$ and so $\overline{Y}_{|K_{i,j}|} \to p$, or, equivalently, that

$$\hat{P}(Y = 1|X_i = j, H_1^i) \to P(Y = 1|X_i = j) \tag{3.4.1}$$

As for the probability of the association being real, $\hat{P}(H_1^i)$, it will be used without formal proof that for those loci $i \in J$ for which the underlying null hypothesis of no association is false, that $\hat{P}(H_1^i) \to 1$ and $\hat{P}(H_0^i) \to 0$ as $n \to \infty$. Similarly, for those loci $i \in I$ where the underlying null hypothesis of no association is true, $\hat{P}(H_0^i) \to 1$ and $\hat{P}(H_1^i) \to 0$ as $n \to \infty$. Note that this is due to the definition of $\hat{P}(H_0^i)$ given in Equation (3.3.3), which we restate here as

$$\hat{P}(H_0^i) = \frac{P(H_0)P(T^i|H_0)}{P(H_0)P(T^i|H_0) + P(H_1)\int_\theta P(T^i|\theta)P(\theta|H_1)d\theta}$$

where the parameter $\theta = (RR_1, RR_2)$. As the sample size increases to infinity, then for those loci in which the null hypothesis is true, that is $RR_1 = RR_2 = 1$, then this should force the probability under the alternative hypothesis to 0, which forces $\hat{P}(H_0^i)$ to 1. Similarly, for those SNPs which the null hypothesis is false, this should force $P(T^i|H_0)$ to 0 as the sample size goes to infinity, making $\hat{P}(H_0^i)$ go to 0.

Finally, consider the estimation of $\hat{P}(Y = 1)$ under the two models proposed earlier; (3.3.1), a simple count $\hat{P}(Y = 1) = \frac{1}{n}\sum_{i=1}^{n} y_i$ and (3.3.2), a geometric mean $\hat{P}(Y = 1) = \left(\prod_{i=1}^{m} \frac{n_{1,+}^i}{n^i}\right)^{\frac{1}{m}}$. Under the situation where there are no genotype errors, then the two versions are equivalent;

$$\left(\prod_{i=1}^{m} \frac{n_{1,+}^i}{n^i}\right)^{\frac{1}{m}} = \left(\prod_{i=1}^{m} \frac{n_{1,+}}{n}\right)^{\frac{1}{m}} = \frac{n_{1,+}}{n} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

The righthand side of the above equation reflects the average of the class labels, taken

with respect to an underlying probability $p = P(Y = 1)$. Therefore, using the Strong Law of Large Numbers, as $n \to \infty$, then $\overline{Y}_n \to p$, and so $\hat{P}(Y = 1) = \overline{Y}_n \to p = P(Y = 1)$.

Therefore, as $n \to \infty$, the genotype Naive Bayes Score (3.3.5) function will converge to

$$\begin{aligned}
&(m-1)\log\left(\frac{\hat{P}(Y=0)}{\hat{P}(Y=1)}\right) \\
&+\sum_{i\in I}\log\left(\frac{1}{1-\hat{P}(Y=1)1-\frac{n^i_{1,j}}{n^i_{+,j}}0}-1\right) \\
&+\sum_{i\in J}\log\left(\frac{1}{1-\hat{P}(Y=1)0-\hat{P}(Y=1|X_i=j,H^i_1)1}-1\right) \\
&=\log\left(\left(\frac{P(Y=0)}{P(Y=1)}\right)^{m-1}\left(\frac{P(Y=1)}{P(Y=0)}\right)^{|I|}\left(\prod_{j\in J}\frac{P(Y=1|X_i=j)}{P(Y=0|X_i=j)}\right)\right) \\
&=\log\left(\left(\frac{P(Y=0)}{P(Y=1)}\right)^{|J|-1}\left(\prod_{j\in J}\frac{P(Y=1|X_i=j)}{P(Y=0|X_i=j)}\right)\right) \qquad (3.4.2)
\end{aligned}$$

From the above calculations, we have $nbs_{genotype} \to$(3.4.2). Now, the inside of (3.4.2) is equivalent to (3.2.9) for $|J|$ variables, which was shown in the Theoretical Basis of Naive Bayes Classifier Section 3.2.1 to be equivalent with Equation (3.2.1). Therefore, we can rewrite (3.4.2) as

$$\begin{aligned}
&\log\left(\left(\frac{P(Y=0)}{P(Y=1)}\right)^{|J|-1}\left(\prod_{j\in J}\frac{P(Y=1|X_i=j)}{P(Y=0|X_i=j)}\right)\right) \\
&=\log\left(\frac{P(Y=1|X=x)}{P(Y=0|X=x)}\right)=\log\left(\frac{\eta(x)}{1-\eta(x)}\right)
\end{aligned}$$

Therefore, $nbs_{genotype}(x) \rightarrow \log\left(\frac{\eta(x)}{1-\eta(x)}\right)$, and so

$$\begin{aligned}
&\{x | nbs_{genotype}(x) \geq 0\} \\
&\rightarrow \left\{x \middle| \log\left(\frac{\eta(x)}{1-\eta(x)}\right) \geq 0\right\} \\
&= \left\{x \middle| \frac{\eta(x)}{1-\eta(x)} \geq 1\right\} \\
&= \left\{x \middle| \eta(x) \geq \frac{1}{2}\right\}
\end{aligned}$$

which ensures that as $n \rightarrow \infty$, that $nbs_{genotype} \xrightarrow{\text{a.s.}} g^*$, the Naive Bayes classifier, as required. ∎

In the above theorem, the genotype version of the modified Naive Bayes algorithm was shown to be strongly consistent to optimal Bayes Decision Rule. It seems likely, and it will be conjectured, that the additive modified Naive Bayes algorithm is strongly consistent under an additional assumption that $\eta(x)$ follows an additive model of inheritance. Note that the genotype modified Naive Bayes algorithm would already be strongly consistent for this type of inheritance, as it in the proof of its consistency, no assumptions other than independence were made with respect to the underlying probability distribution.

Chapter 4

Programs

The purpose of the Background chapter was to introduce the necessary statistical and conceptual ideas needed to understand the traditional statistical analysis of a genome wide association study. The purpose of the Methods chapter was to build on the Background chapter and explain how a particular classification algorithm, the Naive Bayes algorithm, could be trained using genetic data. Since many of the procedures explained in the Background and Methods sections are not available either in a form which can handle the size of the dataset under analysis, or are new methods, a significant amount of code had to be generated for this thesis *de novo*. The purpose of this chapter is to present an overview of the computer programs which were written for this thesis, and to describe some of the computational issues which arose during this thesis.

This chapter will be organized as follows. First, a summary of the programs which were written during the course of this thesis will be given as a flowchart in Figure 4.1 and as a brief written summary. The written summary will include references to the appropriate Background and Methods sections which were required in coding those sections. Finally, in the second section, a short description of the computational software and hardware used will be given. Note that selected pieces

of code representing the core issues of the thesis will be presented in Appendix B.

4.1 Overview of Processes

Given the large number of steps necessary to prepare the data for analysis, and then run the analysis itself, processes were partitioned into manageable subprojects, which were run independently and sequentially of one another. The steps are listed in the flowchart depicted in Figure 4.1, and described in the text in the list which follows. The code for steps 4, 6 and A, representing the r^2 filter, Naive Bayes and posterior probability calculations, are provided in an Appendix in Section B.2, Section B.3, and Section B.4, respectively. Note that the code for Section 6 will omit the graphing code which was used to make Figures in the Application chapter.

1. **Input Data** Input Ottawa Heart Genomics Study, Cleaveland Clinic Foundation, Wellcome Trust Case Control Consortium into R from respective original file formats.

2. **Quality Control** Generate quality control measures for each SNP for each of the three cohorts. This includes the use of the Hardy-Weinberg Equilibrium from Section 2.3.2.

3. **Filter and Merge** Generate summary data for all SNPs passing quality control for each cohort.

4. r^2 **Filter** Apply the r^2 filter to the SNPs which pass quality control in OHGS. The program is modelled after the description of the r^2 filter in the Methods, Section 3.1, and uses the linkage calculations described in Section 2.3.3.

5. **Merge Data** Make a merged genotype file for each cohort with data from SNPs which pass quality control and OHGS r^2 filter.

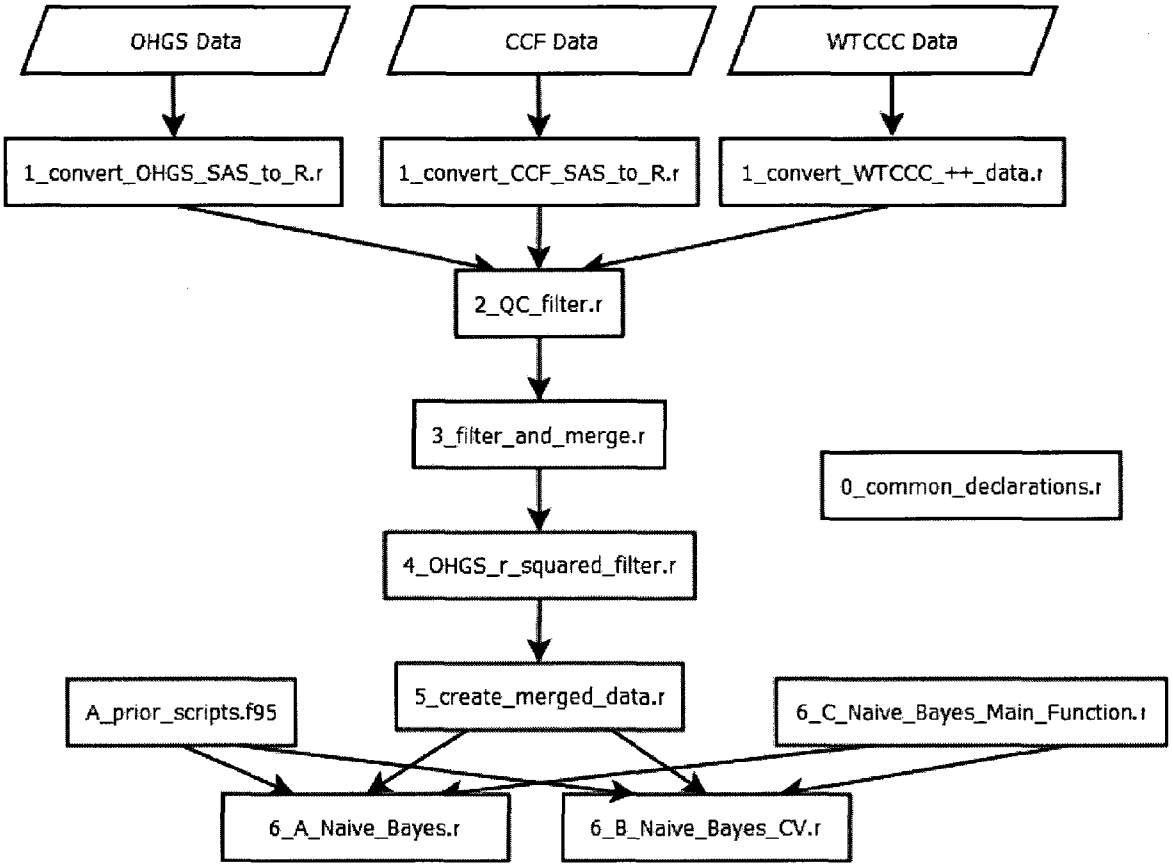

Figure 4.1: Flowchart of Computer Processes. Flowchart lists programs which were written in R (prefixed by numbers) and Fortran (prefixed by letters) to execute the mathematical computations of this thesis. Input is given as parallelograms, programs are given in rectangles, with data passing being represented by arrows. The program `0_common_declarations.r` was used to create a file with variables common to all programs and was invoked by programs 2 through 6. QC=Quality Control, OHGS=Ottawa Heart Genomics Study, WTCCC=Wellcome Trust Case-Control Consortium, CCF=Cleveland Clinic Foundation. This flowchart was created with the use of Dia `http://dia-installer.de`

6. **Naive Bayes** Construct the Naive Bayes classifier using OHGS data and then run on test sets. Run cross-validation to get training set estimates of performance. Graph results. Running the Naive Bayes algorithm is largely based on the description in the Methods, Sections 3.2 and 3.3, which itself uses the Cochran Armitage Test for Trend derivations of Section 2.3.4, the Chi-Square Test derivations of Section 2.3.4, the population genetics of Section 2.3.5 and the posterior probabilities of Section 2.3.6. Note that the calculation of the posterior probabilities, given its computational requirements, was carried out by a Fortran subroutine run through R.

4.2 Computational Specifics

Almost all code was run in R [38], with a small amount being passed as subroutines to Fortran 95. Within R, the Genetics package [49] was used, as well as some code for the Cochran-Armitage Test for Trend which was available at `http://finzi.psych.upenn.edu/R/Rhelp02a/archive/20396.html`. Fortran code was compiled with g95 `http://www.g95.org`.

Calculations were carried out on two desktop computers: computer 1, a 2.8 GHz Pentium D Dual Core with 2 Gb of RAM and an IDE Hard drive; and computer 2, a 3.0 GHz Pentium D Dual Core with 3 Gb of RAM and a SATA Hard drive. Note that the SATA vs. IDE hard drive on computer 2 versus 1 provided a significant performance boost in the early stages of analysis, where several gigabytes of data were read and written to the hard drive repeatedly. Several calculations were also restricted to computer 2, as loading the OHGS merged dataset, with 237602 SNPs, 2997 subjects, and storing data in the raw format (8 bits per cell) caused the OHGS data matrix to be about 680 Mb in size. As R requires that vectors are allocated continuously in memory, this meant that calculations were more easily performed on computer 2.

Running the programs themselves was relatively speedy. The slowest component was generating the posterior probabilities during the construction of the cross-validation Naive Bayes classifiers. Even with the Fortran programs, which were approximately 200 times faster than their non-optimized R counterparts, generating the posterior probabilities for 200000 SNPs took about an hour; hence X-fold cross-validation would take roughly X hours on a single core. Note that integrals in the posterior probabilities were approximated by subdiving the integral into equally sized bins, where 1000 bins were taken for the Additive model, and 10000 bins were taken for the Genotype model (100 per direction).

Chapter 5

Application

The purpose of the earlier chapters was to introduce the relevant statistical and genetic concepts which are necessary to understand the application of a classification algorithm to a genome wide association study. In this section, we use the software from the Programs chapter and data from three cardiovascular GWAs to construct Naive Bayes estimators.

This chapter will be organized as follows. First, the three cardiovascular GWAs which serve as the basis for this thesis will be introduced. Next, the results of applying the quality control filters used in this thesis will be discussed, as well as the results from filtering the data using the r^2 filter described in the Methods chapter. Finally, the results from the various Naive Bayes algorithms will be given. This includes a brief comparison of the posterior probablities with the regular p-values, and the summary results of the classifiers based on cross-validation and test-set results.

5.1 Epidemiological Details of Study Cohorts Used

In this section, we introduce three GWA studies which have been performed. Epidemiological details of these studies are available in Table 5.1. Specific study details

	OHGS Case	OHGS Control	WTCCC Case	WTCCC Control	CCF Case	CCF Control
Number	1542	1455	1926	2938	1603	317
Age at Onset	48.7±7.3	-	49.7±7.7	-	48.6±7.3	-
Age at Consent	-	75.0±5.0	60.1±8.1	-	58.8±10.3	73.2±5.3
Male %	75.9	52.0	79.3	49	75.2	54.9
Body Mass Index	28.5±4.9	26.1±4.0	-	-	29.2±5.5	27.7±5.5
Obese (BMI>30) %	30.4	14.5	23.3	-	37.2	5.1
Diabetes Mellitus %	0.1	7.2	11.0	-	0	0
Hypertension % *	47.5	36.1	42.7	-	71.9	64.3
Smoke Ever %	71.8	46.5	75.8	-	75.0	56.4
Smoke Current %	20.6	2.3	-	-	19.4	3.8

Table 5.1: Results of Clinical Parameters for Study Populations. Note that information on the WTCCC cohort was constructed without a subject level phenotype file, but from information from published GWAs [51, 41]. OHGS=Ottawa Heart Genomics Study, WTCCC=Wellcome Trust Case-Control Consortium, CCF=Cleveland Clinic Foundation.
* OHGS values based on $n_{case} = 1391, n_{control} = 1454$

and inclusion criteria are available in the subsections below. Note that the descriptions given in this section are those of processes which had already been run before this thesis was undertaken, with the exception of some of the summarization given in Table 5.1. Therefore, although the material presented in this section gives 'results', they are not results of this thesis. However, all other results given in this chapter were derived for the purposes of this thesis.

5.1.1 Ottawa Heart Genomics Study

The Ottawa Heart Genomics Study (OHGS) is an ongoing GWA based out of the University of Ottawa Heart Institute (UOHI). In 2007, McPherson *et al.* published the first set of results from the OHGS in *Science*, which showed an association between Coronary Heart Disease (CHD) and a locus on chromosome 9, known as 9p21 [31]. This locus has since been replicated in numerous other studies of varying ethnicities

[51, 41]. Further details on specific loci associated with CHD will be discussed in Section 6.1.

Methodologically, the paper by McPherson *et al.* included 3 cohorts from the Ottawa population, titled the Ottawa Heart Studies 1-3. These three studies contained 322, 311 and 647 cases, respectively, and 312, 326 and 847 controls, respectively. These three studies were later combined with other Caucasian samples from the Ottawa population to form the Ottawa Heart Genomics Study. Currently, there are two OHGSs; a completed OHGS-1, and an ongoing OHGS-2. For the purposes of this thesis, we will focus our attention on OHGS-1, and refer to it simply as the OHGS.

When completed, the OHGS featured GeneChip level data for 1606 cases and 1472 controls, which was later reduced to 1542 cases and 1455 controls through principal component analysis to remove people with non-Caucasian ancestry. Of these, 948 cases and 1017 controls were genotyped with the Affymetrix 500K GeneChip, and the rest with the Affymetrix 1M GeneChip. Inclusion criteria for cases was set as having had either a myocardial infarction, coronary revascularization (coronary angioplasty/percutaneous coronary intervention or coronary artery bypass graft) or a coronary angiogram showing stenosis of at least 50% in at least one coronary artery. Age limits for cases were originally set at ≤ 55 for men and ≤ 65 for women; however, several cases were included which did not meet this criteria: 22 men aged ≥ 56, 4 women aged ≥ 66, 6 men of unknown age at onset, 4 women of unknown age at onset. Cases were generally excluded if they had diabetes mellitus or overt hyperlipidemia; however, 1 case was included for whom additional phenotypic information later revealed the presence of diabetes at the onset of cardiovascular disease. Two sets of inclusion criteria were set up for controls, due to different acquisition protocols. One set of controls were recruited with inclusion criteria set as being healthy and to have a lack of cardiovascular disease history. For the other set of controls, recruited through the catheterization lab at the UOHI, inclusion criteria was set as having an angiogram which showed that none of the coronary arteries had a stenosis encompassing greater

than 50% of the vessel. Age cutoffs for controls were originally set as men aged 65 and older, women 70 and older; however, several controls were included which did not meet this criteria: 5 men aged ≤ 64 and 30 women aged ≤ 69.

More details on the OHGS population can be found in a recent paper from the UOHI by Stewart *et al.* [45], although phenotypic information in the paper pertaining to cutoffs does not reflect the up to date phenotype information for the population being studied.

5.1.2 Wellcome Trust Case Control Consortium

The Wellcome Trust Case Control Consortium (WTCCC) published in 2007 a genome wide association study using a combined set of 3000 controls against 2000 cases each for 7 major diseases, among them Myocardial Infarction (MI)/Coronary Artery Disease (CAD) [51]. Included in the original publication was a wealth of new loci for these diseases, along with new statistical methods and software.

For the CAD/MI analysis in the WTCCC, initial genotyped case and control populations of 1988 and 3004 were reduced to 1926 and 2938, respectively, through a method similar to principal components analysis, which uses an identical-by-state matrix instead of a correlation matrix, and multi-dimensional scaling instead of principal components analysis. More details are available in their publication [51]. All cases and controls were genotyped using the Affymetrix 5.0 array. Cases were defined as having had either a myocardial infarction or coronary revascularization (coronary angioplasty or coronary artery bypass graft) before the age of 66. Controls were not screened for absence or likely absence of cardiovascular disease, nor was an age cutoff used; instead two representative population cohorts were taken from Britain. The first cohort, the 1958 Birth Cohort (58C), is a subset of all births which occurred in Britain during a particular week in 1958 for which ethical consent and DNA were available, selected to be geographically representative of Britain. The second cohort,

from the UK (National) Blood Services (NBS), was formed from a subset of ethically approved anonymous blood donors in Britain, taken to be representative of geographic variation and age of Britain. The two controls populations were composed of 1500 (58C) and 1504 (NBS) samples, and formed the original set of 3004 controls under investigation. More details on the epidemiological nature of the WTCCC cohorts can be found in the 2007 *Nature* publication [51].

5.1.3 Cleveland Clinic Foundation

The Cleveland Clinic Foundation (CCF) has only recently completed a GWA for coronary artery disease. Most cases included had a $> 50\%$ stenosis in at least one coronary artery, as determined by coronary angiography. All CCF samples were genotyped with the Affymetrix 1M GeneChip at the University of Ottawa Heart Institute. Age limits for onset of disease in cases was set at ≤ 55 for men and ≤ 65 for women. Age cutoffs for controls was generally set as 65; however 8 male controls were included of age between 64 and 65.

5.2 Quality Control

The process of Quality Control (QC) is not an easy one, especially in a genome wide association study. QC procedures were employed sequentially and independently; the details are summarized in Table 5.2. This section will include a brief description of each step, as well as the rationale for their inclusion.

Due to the fact that two DNA microarrays were used, the Affymetrix 500K and 1M, the first procedure was to harmonize the set of SNPs to be analyzed. Since the primary dataset of interest, the OHGS, was composed of a mixture of 500K and 1M genotypes, an intersecting subset of $482,251$ SNPs common to both the 500K and 1M chip was selected for further analysis. All three datasets, the OHGS, WTCCC

and CCF, had genotype level data for all of these SNPs.

The next step was to remove X chromosome SNPs. Although interesting, the analysis of X chromosome SNPs is made more difficult due to their unequal inheritance in men and women; women have two copies of the X chromosome while men only have one. As the 10,212 X chromosome SNPs make up only a small percentage of the 482,251 SNPs (2.1%), they were removed.

Following the removal of the X chromosome SNPs, a Call Rate (CR) and Minor Allele Frequency (MAF) filter was used. Recall that the CR is the proportion of samples called a non-null genotype by the genotype calling algorithm at a SNP, and the MAF is the proportion of the less common allele for the SNP. The general rationale behind having a CR filter is to remove SNPs with a low CR, as a larger proportion of these SNPs are not being called confidently by the calling algorithm. The rational behind entangling the removal of SNPs based on CR and MAF is to set a more stringent CR for SNPs with a low MAF. Specifically, this helps ameliorate two problems, namely that SNPs with a low MAF are more likely to generate spurious associations (Type I errors), and SNPs with low MAF are more likely to be called incorrectly due to issues surrounding clustering rare genotypes. To this end, SNPs were excluded which met the following criteria

$$((CR < 95\%)\&(MAF \geq 5\%)\, or\, ((CR < 99\%)\&(5\% > MAF \geq 0\%))$$

In other words, all SNPs were removed with a $CR < 95\%$, and SNPs with a MAF of less than 5% had to meet a 99% CR. This filter was harmonized with the one used by the WTCCC [51].

Following the CR/MAF filter, a Hardy-Weinberg Equilibrium (HWE) filter was used. Recall from Section 2.3.2 that testing for HWE tests as to how well the observed proportions of homozygotes and heterozygotes fit with the expected proportions assuming that the locus is in Hardy-Weinberg Equilibrium in the control population.

As all SNPs are assumed to be in HWE in the control population, then SNPs with low HWE p-values are likely the result of genotyping errors, either due to contamination of a batch of DNA samples or through improper clustering in the genotype calling algorithms. To ensure removal of these SNPs, the control results for all SNPs were tested using the Exact form of the HWE test previously introduced, and SNPs were removed if their p-value was less than the $p_{threshold} = 5.7 \times 10^{-7}$. Note that this is a liberal cutoff, in the sense that only the most egregious violations of HWE are removed, and it is unlikely a SNP which has been properly genotyped will be removed by chance. This value is based on empirical data from the WTCCC, and was harmonized with the filter they used [51].

Following the HWE filter, a filter was used to remove two potential sources of error which could arise specific to certain studies. For the OHGS, this potential source of error lies with the fact that two different platforms, the Affymetrix 500K and 1M GeneChip were used; for the WTCCC, the potential source of error lied with the fact that two different control groups were used. Therefore, for the OHGS control population genotyped using the 500k vs. 1M, and the WTCCC control population NBS vs. 58C, the following filter was applied. SNPs were removed if the p-value from either a 1 or 2 degree of freedom chi-square test or a 1 degree of freedom Cochran-Armitage Test for Trend taken between the two control population subgroups was lower than a threshold, set at $p_{threshold} = 5.7 \times 10^{-7}$. Note that a 1 degree of freedom chi-square test corresponds to SNPs with a low MAF where only two genotypes are present. Again, this filter was harmonized with that used by the WTCCC [51].

Following the removal of plate/population differences, one more MAF filter was applied, this time to remove SNPs with a study-wide MAF of less than 1%. This was done for several reasons: as low frequency SNPs often give spurious results, and as low frequency SNPs take up a lot of computer memory for a small incremental gain. Note that this filter was sometimes applied in the WTCCC paper; it was not applied in the general analysis of results, but was applied for procedures like

multidimensional scaling and visualization of results. We chose to implement this filter for all analyses for the reasons given above, but left its implementation to follow the more traditional filters so that comparisons can be made as to how many SNPs fail each of the QC filters for the condensed 482,251 dataset as compared to with in the WTCCC's original 500,568 SNP dataset. Note that since SNPs are only taken which pass all filters, the order in which filters are applied is irrelevant.

Lastly, the WTCCC had one more filter, which was applied in a post-hoc fashion to any SNP which showed evidence of association with disease. For these SNPs, visual inspection was made of the cluster plots corresponding to genotype calls, and SNPs were removed if clustering was deemed unsatisfactory, such as if two clusters overlapped and there was a bias as the null genotype calls, or if one cluster was missing for either the case or control group. This filter was not applied to the OHGS or CCF datasets for this study. Note that this topic will be discussed in more detail in the Discussion.

Following the application of these filters, the OHGS retained 364,015 SNPs for further analysis, the WTCCC retained 388,731 SNPs and the CCF 378,591 SNPs.

5.3 r^2 filter

As was described earlier in the Methods, Section 3.1, an r^2 filter was used to thin the set of all SNPs into a subset of SNPs, such that for any SNP i in the set of all SNPs M, there exists a SNP j in the subset $M_{r^2_{threshold}}$, such that $r^2(i,j) \geq r^2_{threshold}$ (Note that i can equal j). Recalling that r^2 is a measure of one SNP's ability to tag another, the purpose of this filter was to remove those SNPs which were redundant, in the sense that they were nearly identical to other SNPs on the GeneChip which had been genotyped. This reduces the computational burden of data analysis and increases the independence between SNPs, which itself increases power in traditional analyses and lessens the effect of non-independence on algorithms which work on the

	OHGS	WTCCC	CCF
Original number of SNPs	482251	500568	909622
Synchronized SNP set	482251	482251	482251
Remove X Chromosome SNPs	10212	10212	10212
Remove $(CR < 95\%)$ & $(MAF \geq 5\%)$ or $(CR < 99\%)$ & $(5 > MAF \geq 0\%)$	52623	20101	45175
Remove HWE $p < 5.7\times10^{-7}$ in controls	19334	3520	119
Remove Chip/Control group $p < 5.7 \times 10^{-7}$ *	1191	60	N/A
Remove $MAF < 1\%$	34876	59300	48154
Remove Bad Clustering	-	327	-
Remaining	364015	388731	378591

Table 5.2: Results of applying Quality Control Filters. Note that the the numbers correspond to SNPs removed as the filters were employed sequentially top to bottom. OHGS=Ottawa Heart Genomics Study, WTCCC=Wellcome Trust Case-Control Consortium, CCF=Cleveland Clinic Foundation, HWE=Hardy-Weinberg Equilibrium, CR=Call Rate, MAF=Minor Allele Frequency.
* Chip/Control p values are the minimum of tests of association (1 df Cochran-Armitage Trend Test or 1 or 2 df Chi Square Test) between the two GeneChip types (OHGS) and control types (WTCCC)

basis of independence, including, for example, the Naive Bayes algorithm.

We applied the r^2 filter to those SNPs which passed the QC filters of the previous section; specifically, we applied it to the OHGS control dataset, which began with $364,015$ SNPs, using an $r^2_{threshold} = 0.8$. We chose the threshold of 0.8 as this is a relatively stringent threshold that ensures SNPs are highly related before removal. Note that others have previously used this threshold for similar procedures, for example Carlson *et al.* used an $r^2_{threshold} = 0.8$ in their paper on selecting a maximally informative set of SNPs for assay design [11].

With the application of the r^2 filter, $126,413$ largely non-informative SNPs were removed, leaving $237,602$ SNPs which passed both the QC and r^2 filter from the OHGS dataset. Filtering the WTCCC dataset to include only those SNPs which passed both its QC and the OHGS r^2 filter left $231,773$ SNPs; for the CCF $228,859$. See Table 5.3 for the full results.

	OHGS	WTCCC	CCF
Initial SNPs	364015	388731	378591
Remove filter $r^2_{threshold} = 0.8$	126413		
Remaining	237602		
Concordant	237602	231771	228859
Percent Concordant	100	97.5	96.3

Table 5.3: Results of applying r^2 Filter. Note that the r^2 filter is applied to the OHGS study only; the SNPs from the WTCCC and CCF datasets used for further analysis must pass both their respective QC filter and be included in the list of SNPs which pass the OHGS r^2 filter. Note that concordant refers to SNPs in a particular study which both passed QC in their study and passed the r^2 filter in the OHGS. OHGS=Ottawa Heart Genomics Study, WTCCC=Wellcome Trust Case-Control Consortium, CCF=Cleveland Clinic Foundation.

5.4 Results from Naive Bayes

Following the application of the various filters, the data was finally of a sufficient quality to allow for the process of constructing the various Naive Bayes classifiers, according to the methods laid out in the Methods chapter.

Note that this process involved the determination of posterior probabilities for each SNP which passed quality control and the r^2 filter. For the purposes of this thesis, we tried two different priors: a 'many' prior where $P(H_0) = 1 - \frac{1}{2000}$; and a 'few' prior where $P(H_0) = 1 - \frac{1}{20000}$. Note that these priors correspond to an assumption that among 500,000 independent genetic loci, that there are 250 or 25 loci which are associated with coronary artery disease, respectivel.

For reference purposes, two figures have been included which show how the posterior probabilities compare to traditional p-values. Figure 5.1 gives the posterior probabilities for the 100 SNPs with the lowest posterior probabilities for the two inheritance models and two priors. Figure 5.2 plots the best posterior probablities against the most appropriate p-value for each of the two inheritance models and two priors. The results of Figure 5.2 seem to show that there is general agreement between those SNPs which are ranked well by the posterior probability as compared to

the p-values.

As for the results from the classifiers themselves, Figure 5.3 displays a summary of the results generated by training on the OHGS dataset and testing on the WTCCC and CCF datasets. Note that cross-validation results are also given for the OHGS dataset. Results were generated for both the Additive and Genotype models for the modified Naive Bayes classification algorithm using the 'many' and 'few' prior, and also for the traditional Naive Bayes classification algorithm for the two models. It is interesting to note that the OHGS cross-validation results significantly outperform the WTCCC and CCF datasets, which are surprisingly similar to one another, and that none of the classification algorithms seem to outperform each other.

Figure 5.4 shows the actual scores for a particular model for each subject, using the Genotype model with the 'many' correction. Although any difference between the cases and controls in the WTCCC and CCF datasets is quite hard to see, it is easier to see a trend in the OHGS data of cases having higher Naive Bayes Scores than the controls. Interestingly, among the OHGS cases, there is a subgroup of subjects, numbered roughly 1250-1500, which have higher Naive Bayes Scores than the other cases. Upon further phenotype review, it turned out that these cases were composed of two different groups of cases, one of which was phenotypically distinct from the bulk of the cases.

One of these groups of cases, with 149 subjects, came from the Coronary Artery bypass Risk Assessment (CARA) clinic at the University of Ottawa Heart Institute, run by Dr. McPherson (Dr. Ruth McPherson, personal communication). These cases have extreme forms of cardiovascular disease which require coronary bypass surgery at an early age, and usually suffer from multivessel coronary artery disease or have had multiple myocardial infarctions. The second group of cases, a group of 89 cases, were also from the McPherson lab, however, unlike the CARA cases, they were not obviously phenotypically different from the bulk of the cases. Intuitively, one could consider the CARA cohort to be a 'super' case group, likely to be enriched

with genetic factors predisposing to CAD due to the early incidence of CAD in the population.

Upon further review, it turned out that these 238 subjects had been genotyped on three plates. Returning to the method of DNA microarray genotyping, a plate is a small plastic tray, approximately 10 centimeters by 15 centimeters, containing 96 wells arranged 8 by 12, each well capable of holding a sample of subject DNA. During the process of acquiring genotypes from DNA microarrays, subject DNA is processed and plated along with the DNA of 94 other people (and a blank) in a plate. The application of reagents and amplification is then sped up by working on 90+ samples on one plate at a time. Working by plate allows for speedy genotype acquisition and the averaging of reagent volumes across many samples. However, it does lead to the potential for intra vs inter plate differences. Things such as contamination of a plate with an external DNA source, incomplete reagent application or processing time could lead to differences between plates and hence similarity within plates. Furthermore, upon review of the method in which genotypes were acquired, it became apparent that the genotype calling algorithms had been run by plate, not for the whole study at once, and hence were likely to be a source of inter-plate variability particularly with the Birdseed algorithm. This last point will be discussed more in the Discussion in Section 6.3.3.

With that in mind, the cross-validation results for the OHGS were rerun, with the aim of assessing the impact of intra-plate similarity. This was done by exchanging the traditional random 10-fold cross validation approach for one in which the subjects corresponding to one plate were removed one at a time as the removed fold. This form of CV was then repeated with the plate numbers scrambled. The difference between the two forms of cross-validation, one in which all members of a plate are removed versus random removal, should give an estimate as to how much the intra-plate similarity accounted for the increased performance of the OHGS CV versus WTCCC and CCF, as well as the seemingly increased performance of the 238 cases

from OHGS in Figure 5.4. The results are shown in Figure 5.5 and Figure 5.6.

Figure 5.5, much like Figure 5.3, shows the cross-validation results for the two different CV schemes, by plate or random. In all, subjects were acquired on 39 plates, making the cross-validation by plate a 39-fold CV, with a median of 84 samples being on each plate. Note that there are less than 90 subjects on most plates as samples were removed by outlier analysis, later restrictions on phenotypes or because a portion of the plates acquired genotypes were from another study. That being said, the results from Figure 5.5 show a clear difference between the performance of the two CV schemes, with the by plate CV scheme clearly under-performing the random CV. As such, this strongly suggests the presence of intra-plate bias as an influence on the performance of the Naive Bayes classifier.

Figure 5.6 shows the subject level Naive Bayes scores for the two different CV schemes for the Genotype probability inheritance model and the 'many' prior. Note that the cases with elevated Naive Bayes Scores have been blocked off, with the phenotypically non-distinct cases being to the left of the CARA cases, which are to their right. Interestingly, running CV by plate did not completely remove the effect, although it seemed to have diminished it.

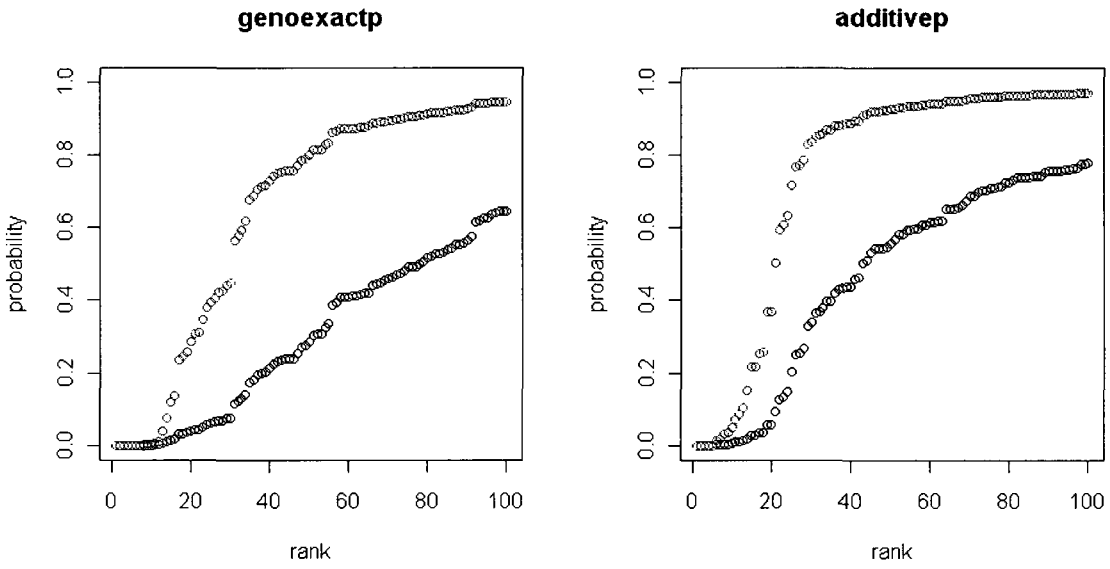

Figure 5.1: Posterior Probability of SNPs with Low Posterior Probabilites. Figure shows the posterior probability of association for the 100 SNPs with the lowest probability of association, plotted by rank. Posterior probabilities are given for the Genotype model on the left, Additive model on the right. Posterior probabilities are given for the 'many' prior (black/dark) and 'few' prior (red/light) for each plot.

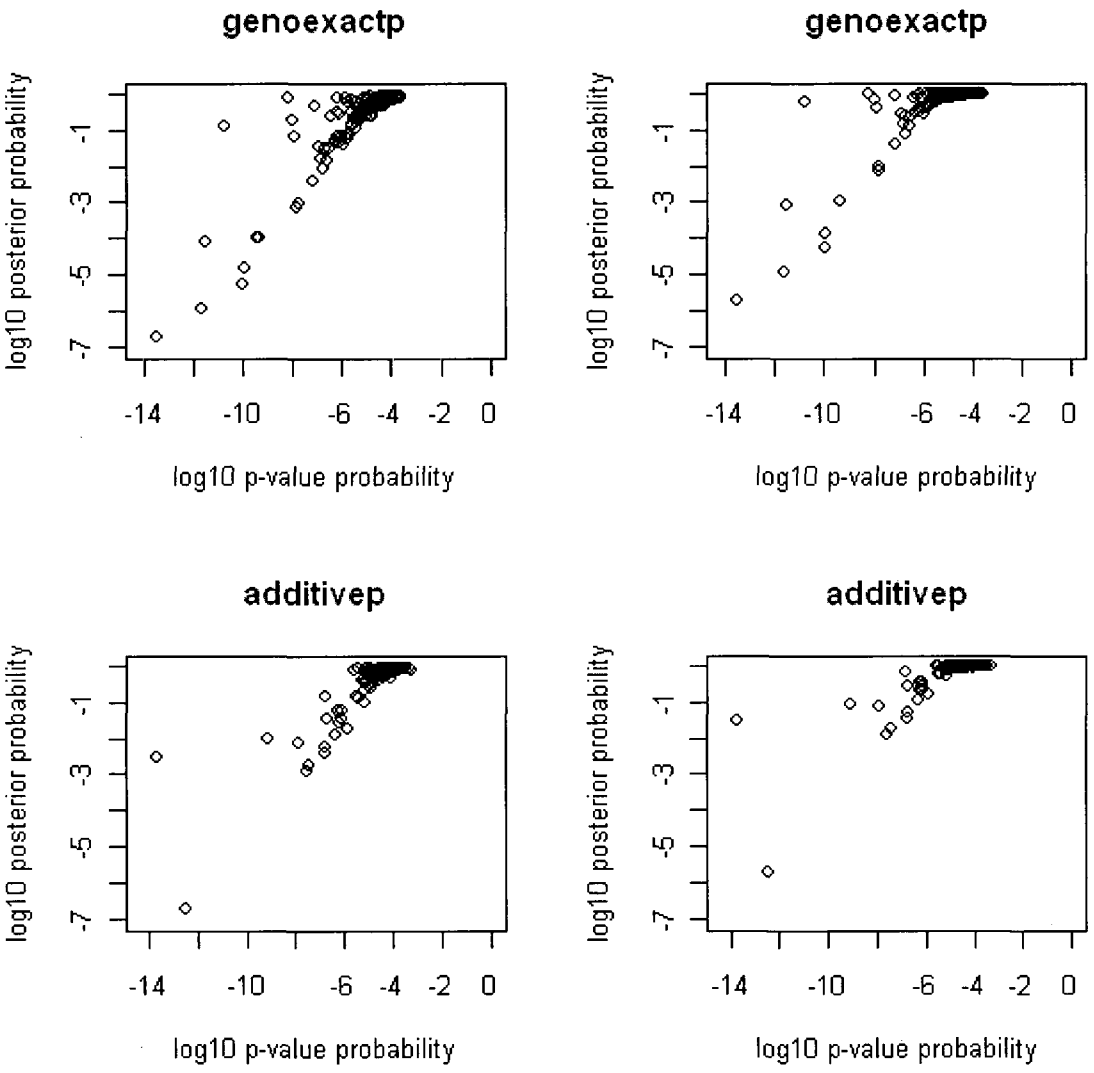

Figure 5.2: Plot of Posterior Probabilities versus p-values. Plots give the logarithm base 10 of the p-value versus the logarithm base 10 of the posterior probability. Plots in the upper row correspond to p-values obtained from Fisher's Exact Test and the posterior probability from the Genotype model. Plots in the lower row correspond to p-values obtained from the Cochran-Armitage Test for Trend versus the posterior probability from the Additive model. Comparisons are made with the posterior probability using the 'many' prior in the left column; the 'few' prior in the right column. Results are given for the 500 SNPs with the lowest posterior probabilities for each model.

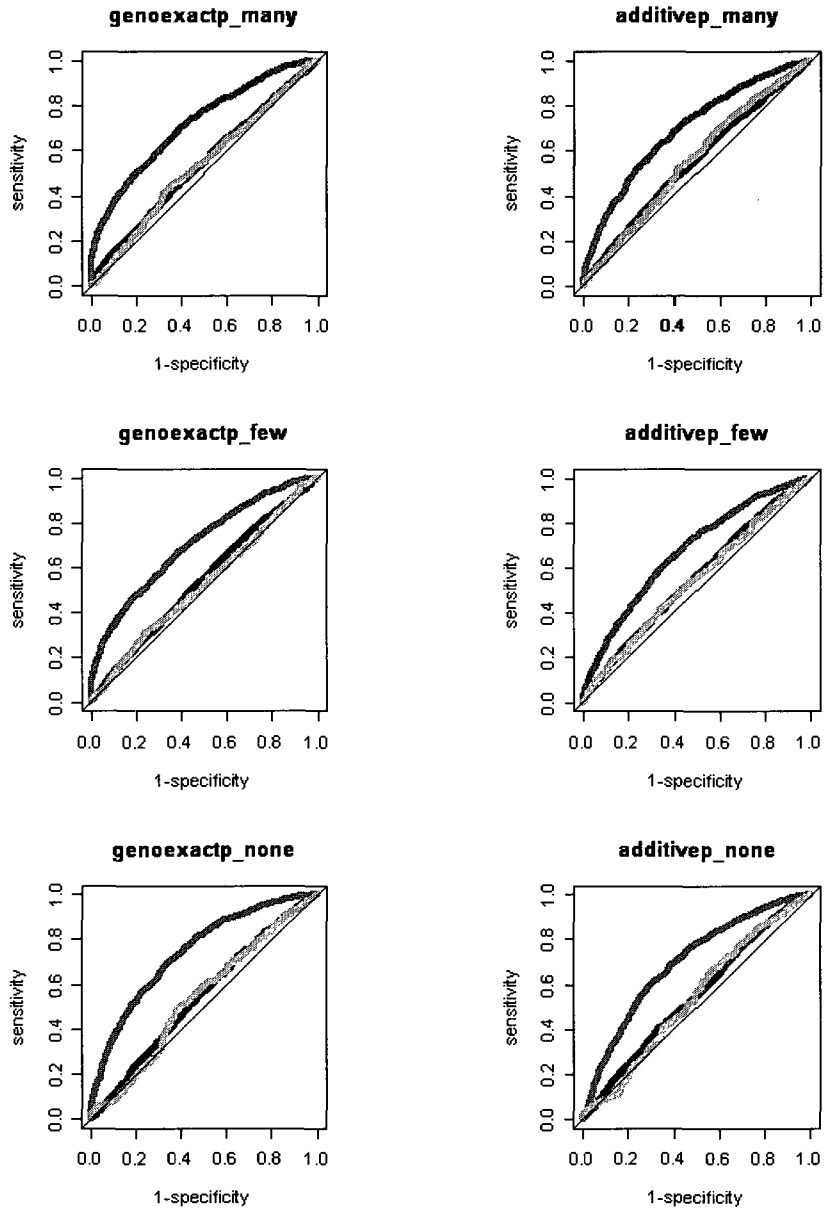

Figure 5.3: Results for the Classification Accuracy of Models on OHGS (Red/Medium Dark), WTCCC (Black/Dark) and CCF Data (Green/Light). Results are shown as receiver operator characteristic curves and display 1 – specificity on the horizontal axis and sensitivity on the vertical axis. Results in the left column correspond to the Genotype model (genoexactp); right column, Additive model (additivep). Results in the top row correspond to the 'many' prior, middle row 'few', bottom row the traditional unmodified Naive Bayes (none).

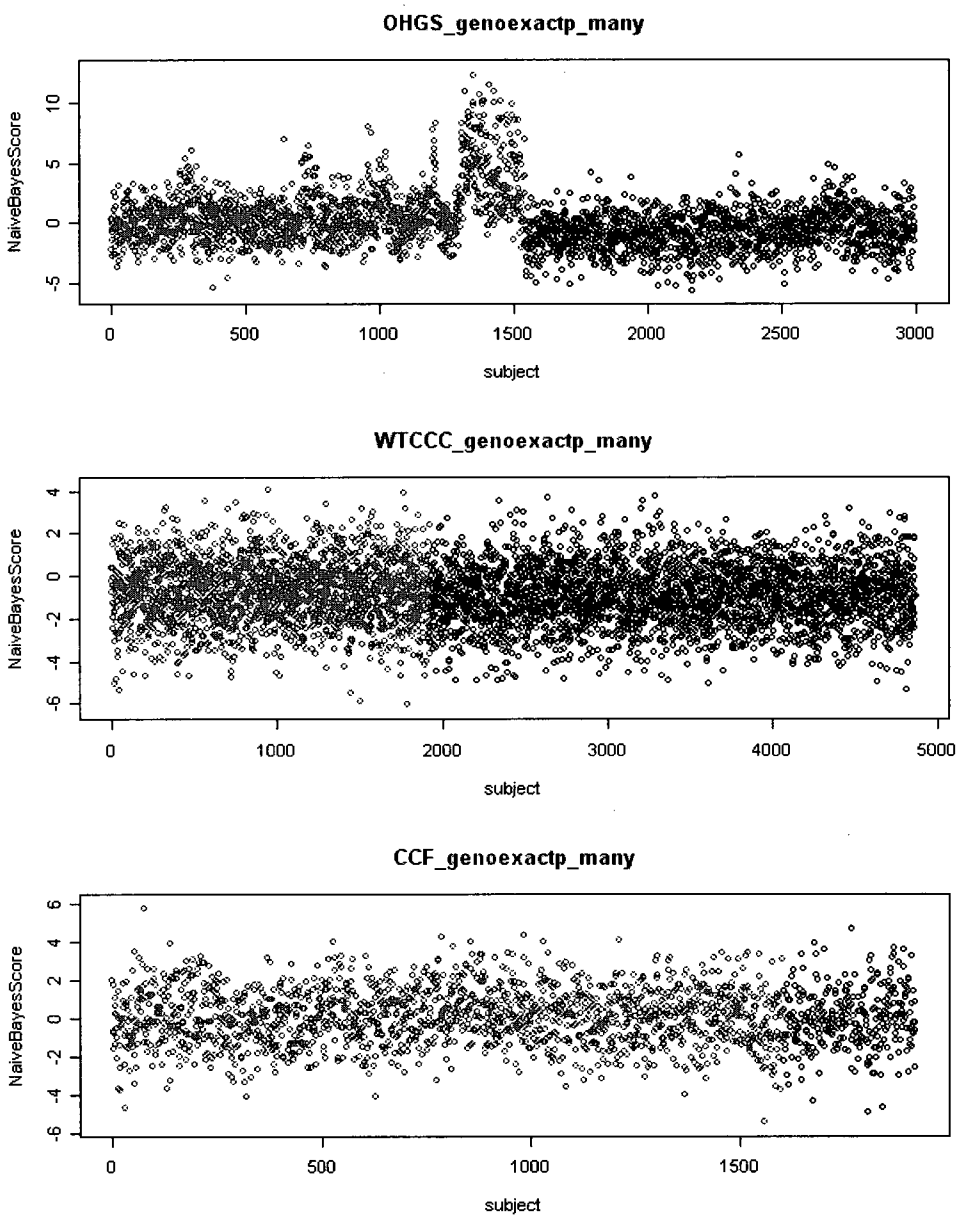

Figure 5.4: Results for Naive Bayes Scores for Genotype Model, 'Many' Prior. Shown are the Naive Bayes Scores for the OHGS (top), WTCCC (middle) and CCF (bottom), obtained using either 10-fold CV (OHGS) or the normally constructed classifier (WTCCC and CCF). Naive Bayes Scores are shown on the vertical axis; individuals are shown left to right on the horizontal axis, with cases being coloured red (light) and controls black (dark).

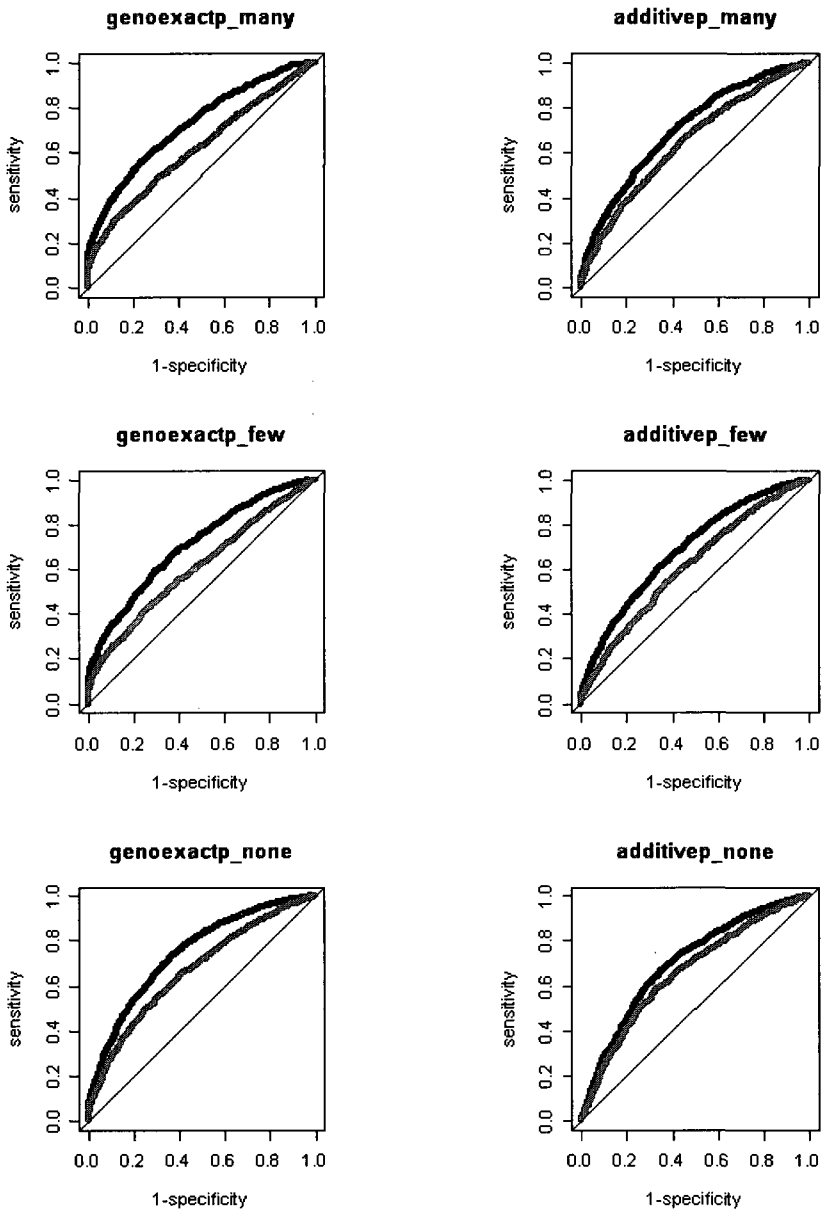

Figure 5.5: Results for Cross-Validation Random versus Plate Specific. Cross-validation performed on OHGS data was 39-fold, being either random (black/dark) or removing the results from one plate (red/light).

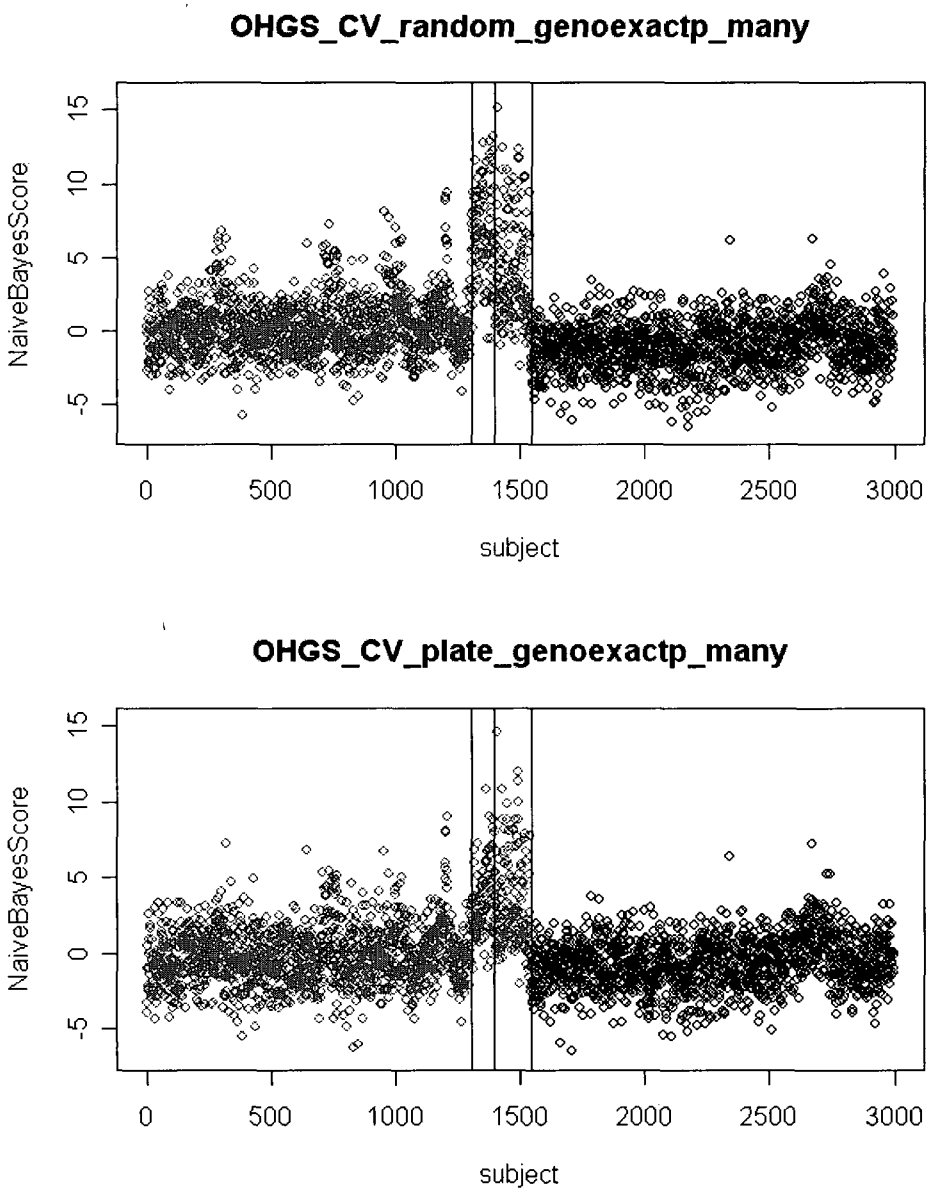

Figure 5.6: Results for Cross-Validation Random versus Plate Specific, Genotype Model, 'Many' Prior. Cross-validation performed on OHGS data was 39-fold, being either random (top) or removing the results from one plate (bottom). Cases are shown in red (light), controls in black (dark). Vertical lines are at subjects 1305, 1394 and 1543, demarcating the non-phenotically distinct case group (1305:1394) from the CARA group (1395:1542).

Chapter 6

Discussion

The purpose of the earlier chapters was to apply a classification algorithm to genome wide association study data. In this chapter, we will discuss the results of the Application chapter, as well as issues which arose during the development of this thesis as to the theory and application of the various methodological steps.

This chapter will be organized as follows. The first section will discuss the current state of cardiovascular GWAs and what loci have been identified. This will prove relevant as the implications and relevance of this thesis are not disease invariant; depending on prior published works, the strategy for constructing classification algorithms from genetic data may change. The next section will deal with methodological and epidemiological issues which arose during the course of this thesis. The third section will deal with the results generated by the r^2 filter, posterior probabilities used in the modified Naive Bayes classifiers, and the observed intra-plate similarity. The fourth section will discuss other classification algorithms which were considered along with the Naive Bayes algorithm. The final section will discuss attempts which have been made to integrate genetic knowledge into the clinical world, and on the future generalizability of classification algorithms generated in a method such as in this thesis.

6.1 The Current State of Cardiovascular GWAs

Although this thesis is of a mathematical nature, it is highly influenced by the results from other published studies dealing with finding variants associated with CAD, even if those studies themselves are not mathematical in nature. Recall that the goal of this thesis was to develop a model to predict CAD from genetic data; however, the landscape in which this thesis was conceived, and the body of knowledge which exists relating to genetic factors predisposing to CAD, has changed significantly during the course of the progression of this thesis. Understanding what has been discovered over the past two years, and the studies which now exist, will help to better understand the rationalization which occurs later on in the Discussion section with regards to the future of creating classification algorithms from GWA data.

Generally speaking, the publishing of CAD GWAs has come in waves. The first wave occurred during the summer of 2007, with the publishing of four papers. The first two papers, by McPherson *et al.* [31] and Helgadottir *et al.* [21], were published back-to-back in *Science* in the June 8th 2007 edition, although the results were available online as of May 3rd 2007. These two papers both independently identified an approximately 58,000 base pair region on Chromosome 9 between 22.06 Mb and 22.12 Mb, in a region known as 9p21, in which a group of tightly linked SNPs (pairwise r^2 of about 0.8 to 1.0) were identified as being associated with CAD. These two papers were followed up by a publication by the Wellcome Trust Case Control Consortium in *Nature* in the June 7 2007 publication [51], which was not available online as early as the *Science* publications (accepted for publication May 11 2007; unsure of online publication date). The paper by the WTCCC also independently found the 9p21 locus, along with finding suggestive associations at several more loci. Shortly afterwards, a fourth paper, by Samani *et al.*, was published in the *New England Journal of Medicine* [41], again finding 9p21 but also strengthening the apparent associations of several suggestive associations found by the Wellcome Trust paper.

Following this first discovery wave of CAD GWA publications, which all identified the 9p21 locus as the most influential genetic locus for CAD but have failed to definitively find any other loci on their own, there was a wave of confirmatory publications for the 9p21 locus. These papers showed that 9p21 was associated with risk in multiple populations, as well as being associated with related but distinct phenotypes, such as intracranial aneurysms and abdominal aortic aneurysms [22] (an aneurysm is a localization blood filled ballooning of the vessel wall). There were also publications which started to confirm some of the suggestive associations found in the WTCCC and Samani papers, namely two loci on chromosome 1, 1p13 and 1q41, along with a locus on chromosome 10, 10q11 [13]. Note that chromosomal regions are often listed X{p/q}Y, to indicate their presence on chromosome X, short (p) or long arm (q) of the chromosome, and where they stain on that arm of the chromosome (11 close to the centromere, higher numbers further away).

The second wave of discovery CAD GWA publications came out in *Nature Genetics* in early 2009, with an issue dedicated principally to publishing results from CAD GWAs. These back-to-back publications identified several new loci, including ones at 3q22 and 12q24 by Erdmann *et al.* [17], ones at 21q22, 6p24, 2q33, 19p13 and 1p32 by the Myocardial Infarction Genomics Consortium (MIGC) [33] and one at 6q26-27 by Trégouët *et al.* [47]. One difference between these publications, and the first wave that came out in the summer of 2007, is that all of these publications required replication to publish. That is, while studies such as the WTCCC published the results of one GWA with 5000 subjects, the papers which came out in *Nature Genetics* all had initial discovery cohorts *and* replication in the same paper. For example, in the MIGC publication, a multi-stage elimination process was carried out with an initial discovery sample of approximately 6000 people, but meta-analysis results were given for a final effective sample size of approximately 19,000.

This is reflective of the general change in GWA publications, which have made built-in replication almost a requirement to publish. That being said, with these

publications, the number of loci associated with the risk of CAD identified through GWAs rose to 12. As of the writing of this thesis during the summer of 2009, these appear to be the only loci definitively associated with CAD identified through GWAs.

6.2 Methodological and Epidemiological Issues

In this section, we will talk about a few methodological and epidemiological issues which arose during the preparation of the data for classifier construction. They are the use of a filter by the WTCCC which we will call a 'bad clustering' filter, and the debate over the nature of the cases and controls which should be included in a genome wide association study.

6.2.1 Bad Clustering Filter from WTCCC

The 'bad clustering' filter from the WTCCC, for lack of a better term, is a quality control filter which was implemented by the WTCCC to remove SNPs. As quoted in the WTCCC paper [51],

> The aim of filtering is to exclude poor SNPs but without removing genuine associations. No single criterion will do this. In order not to exclude possible genuine associations, we chose to apply relatively light quality control filters but then to subject all apparently associated SNPs to visual inspection of cluster plots (See Supplementary Information). Around 100 cluster plots were assessed per disease.

In the end, there were 327 SNPs removed from consideration from analysis in this thesis due to this post-hoc method, although this is fewer than were removed in the original paper due to differences in quality control filters. Although 327 represents a small percentage of the 388371 WTCCC SNPs that eventually passed our filters,

all of these SNPs were analyzed because they showed an apparent association with disease.

This step was not performed for the OHGS or CCF datasets, for several reasons. First, in the case of OHGS, multiple platforms were used to collect genotypes, namely the Affymetrix 500k and 1M chips. As a result, multiple genotype calling algorithms were used, making it difficult to visually inspect results across the groups of cases or controls when the intensities are normalized differently. Second, data analysis for this thesis was performed on already called genotypes; even if the fluorescence intensities were readily available, which they were not, it would be a considerable amount of work to generate the algorithms to inspect cluster plots. Lastly, even though employed by the WTCCC group, this is not a particularly common data quality step in GWAs. The current requirements for publishing GWAs almost always include some sort of replication, in a different population and genotyped in a different way or at a different facility. The purpose of the replication step is to ensure not only that the results were not a statistical artifact, but that they were not the result of some other non-random sampling based event. That is, while quality control filters should catch most genotyping errors, it is not inconceivable that some SNPs will pass quality control that have been very badly genotyped. However, when replication is performed, it is very unlikely that a SNP would appear associated with a trait for both an initial discovery set and an independent replication cohort as the result of errors in *both* populations. This last point emphasizes the importance of replication in the field of GWAs, and the lack of importance in ensuring that one has perfect confidence in every SNP collected in an initial discovery cohort.

With that said, although filters such as the WTCCC 'bad clustering' filter are certainly desirable for the individual study, they do little to help with the process of proving that a locus is associated with a disease, when such proof requires replication. As such, since the primary goal of most GWAs is on the discovery of loci, filters such as the 'bad clustering' approach of the WTCCC are not a high priority. As such,

no such filter was employed in the OHGS or CCF datasets during the traditional statistical analysis which was applied to it.

However, for an analysis such as the one performed in this thesis, such a filter would have proven very useful. For example, in constructing the model with the two different priors, only about 50 to 200 SNPs had a profound effect on classifier construction, depending on the prior and probability model. These SNPs, especially those with very low posterior probabilities, should evoke reasonable confidence that the results are real. However, upon comparison to the WTCCC and CCF datasets, there is little evidence to suggest that these SNPs, with the exception of 9p21, are at all related to CAD (data not shown). In other words, despite the quality control filters employed, there are still a substantial number of results which seem statistically meaningful but are likely spurious. However, given the relatively stringent priors used, especially the 'few' prior, it is likely that these are not Type I statistical errors, but are the result of either an inflation of test statistics from population stratification, or more likely the result of some sort of plate specific genotyping errors which slipped by the quality control filters. This point itself will be discussed more substantially in Section 6.3.3, as this is a very important point of this thesis; that the stringent 'few' prior was supposed to remove all but the most convincingly associated SNPs from classifier construction, but almost all of the SNPs which seemed associated showed very little evidence for association in other datasets. Furthermore, this will be touched on in the Conclusion, as if the removal of SNPs which are most likely the result of genotyping errors cannot reasonably be performed, then any classifier constructed from a classification algorithm on a single GWA dataset is unlikely to generalize to test datasets and the population at large.

6.2.2 Selecting Appropriate Cases and Controls

In the Application section, it could be noticed that the phenotypic descriptions of the cohorts were similar, but not exactly the same. This subsection will be organized as follows. To begin, there will be a brief introduction of the similarities and dissimilarities of the three cohorts used for this thesis, the OHGS, WTCCC and CCF cohorts. Following this, the benefits of screening versus not screening controls will be discussed. Finally, the last part of this subsection will deal with deficiencies of these three cohorts with respect to the ultimate goal of this thesis, classification.

In terms of cases, it is difficult to compare the OHGS, WTCCC and CCF cohorts. Although each case in each cohort has some level of disease, they are not necessarily the same; intuitively speaking, a 55 year old diabetic male who smokes and has a BMI of 30 who has CAD is likely to have a smaller genetic predisposition to CAD than would a 35 year old non-diabetic male who doesn't smoke and has a BMI of 24. That being said, with the limited summary phenotypic information available and displayed in Table 5.1, in terms of traditional risk factors such as hypertension, diabetes, smoking, age and BMI, the case groups of the WTCCC, OHGS and CCF were relatively consistent.

However, unlike the case groups, it is easier to compare the control groups, as controls are more easily defined and hence more amenable to comparison. Each of the three cohorts employed different criteria for controls. In the OHGS, there was a mix of angiographically clean elderly subjects, along with healthy and asymptomatic but angiographically unscreened subjects. Note that an angiogram is a medical procedure for determining narrowing of the coronary arteries, with a clean angiogram indicating a lack of narrowing, and hence a lack of coronary artery disease. Furthermore, note the importance of screening for the absence of coronary artery disease, as currently unpublished data here at the Heart Institute suggests that 30%+ of the healthy asymptomatic elderly population over 70 years of age has some form of latent coronary

artery disease (Dr. Sonny Dandona, personal communication).

As for the WTCCC and CCF cohorts, in the WTCCC, the controls were just a random sample of the population, not screened for the disease nor of a particular age. In the CCF, all controls were angiographically clean and elderly. As such, one could consider the CCF to have used 'hypercontrols', being both angiographically clean and elderly, while the WTCCC used a random sampling of the population and the OHGS used a mix of the two.

The question of whether it is better to use controls drawn randomly from the population or screened for the absence of the disease being analyzed depends on multiple factors, such as other methodological issues of the study, as well as what the desired use of the genetic data is. Controls drawn from the population at large, whether they are drawn randomly or to match an already collected case phenotype distribution, have the following advantages over screened controls: they are less expensive to collect, as no screening must be employed during their acquisition to ensure lack of disease; they are easier to collect, as no rationalization must be employed over selection criteria; and they are interchangeable, and can be used as controls in multiple studies. Conversely, screened population controls are more advantageous than random controls in that they are less likely to suffer from the case phenotype, and hence there is no mixing of cases among the controls, leaving more power to detect underlying genetic associations. Basically, population based controls are easier and cheaper to collect, but will hamper the power of a case-control study.

For the purposes of the analysis in this study, on the construction of a classification algorithm from genetic data, neither the OHGS, WTCCC or CCF are ideal. One of the problems, shared by both the CCF and OHGS, is that cases were pre-screened to ensure absence of diabetes. Note that diabetes is a very potent risk factor for coronary artery disease, and has a strong genetic component. Therefore, its removal in a traditional GWA makes sense, because SNPs which predispose to diabetes would likely be picked up in a GWA for CAD due to the increased risk of developing CAD

with diabetes. However, for classification purposes, this is less than ideal. Classification algorithms should be able to work with confounding variables, particularly if the eventual goal is an attempt at classification based purely on genetic data. The goal is not to predict occurrence of CAD in the absence of diabetes - just CAD itself. As such, having removed diabetes from these two studies makes them slightly less than ideal, and impairs the ability of a classification algorithm to detect CAD.

Furthermore, both the OHGS and WTCCC are 'guilty' of selecting cases with an increased likelihood of being genetically predisposed to CAD. In the case of the WTCCC, most of the cases being studied came from a separate study of families with CAD, although of course only one member of each family was selected for inclusion into the study. This would increase the proportion of people with family history of CAD in the study, and hence the amount of the disease in the study cohort explained by genetic factors. The OHGS is similar in this regard, by having purposefully selected young cases with extreme forms of the disease with as minimal as possible a presence of risk factors, such as the CARA cases explained previously (see Section 5.4). Although this manner of selection is beneficial in an initial discovery cohort, it is less useful for classification purposes, as any results, cross-validated or otherwise, would be based upon the assumption that the subjects used to build the classification algorithm would be representative of the population as a whole. As such, the classification accuracy of the algorithm would likely be overinflated.

This last point, on how the particular selection of cases in some of the studies may overinflate the observed sensitivity and specificity results, is also relevant to the selected controls, however from the opposite perspective. Recall that the WTCCC uses unscreened controls, therefore unlike the cases which are likely to be overenriched with genetic predisposition to disease, the controls from the WTCCC are likely to be underenriched for genetic predisposition to disease, as a proportion of the controls are in fact cases. If the proportion of people aged 70 and over with coronary artery disease really is 30%+, then 30%+ of the random population controls will eventually develop

the case phenotype, making them unsuitable as controls. This means results from the WTCCC are likely to underestimate the classification accuracy of the Naive Bayes algorithm, when compared to future applications of the classifier on the population.

In conclusion, none of the OHGS, WTCCC or CCF are perfect, although the OHGS and CCF cohorts are more similarly defined and hence more amenable to joint analysis. Although many of the theoretical considerations made in this thesis were done with assumptions as to the selection nature of the cases and controls, in reality those assumptions were not met, and all results from this thesis with regards to the accuracy of the classification algorithm must be considered with respect to the studies they have been derived from.

6.3 The Results of the Application

In this section, we will discuss the results from the Application section, except for the phenotypic description of the cohorts, which was largely covered in Section 6.2.2. This section will focus on how various aspects of the methods affected the classification results, namely: the application of the r^2 filter; the posterior probabilities used by the modified Naive Bayes algorithm; and the intra-plate similarity.

6.3.1 r^2 Filter

The r^2 filter, introduced in the Methods section, is a relatively non-standard filter for a genome wide association study, despite the benefits explained earlier in Section 3.1. Nonetheless, it is a useful first step before the application of the Naive Bayes classifier, which assumes independence between SNPs.

The results of the filter, displayed in Table 5.3, demonstrate how much of the Affymetrix genotype data is redundant. We were able to decrease the number of SNPs from 364,015 to 237,602, a decrease of 126,413 SNPs or 35%, using an r^2 threshold

of 0.8. This removal was relatively consistent chromosome to chromosome (data not shown). If one were using a Bonferonni cutoff to determine significance, filtering the data with the r^2 filter would change the p-value threshold from 1.37×10^{-7} to 2.10×10^{-7}.

However, despite these results being good, they were not great. The change in Bonferonni adjusted p-value threshold, despite seemingly relatively impressive, is not meaningful on an absolute scale. Furthermore, the r^2 filter did not always work quite as well as anticipated in terms of distilling large blocks of related SNPs down to a single SNP. As mentioned earlier, in the original 9p21 region identified by McPherson [31], Helgadottir [21] and others, between base pairs 22.06 MBp and 22.12 MBp on chromosome 9, there are about 10 SNPs on the Affymetrix 500k GeneChip and about 20 SNPs on the Affymetrix 1M GeneChip within that region which all show similar minor allele frequencies and effect sizes. Any association of these SNPs can be presumed to come from one real underlying signal, but so far it is difficult to determine which of these particular SNPs is either the real underlying causative variant or is most highly linked to the real variant. When we employed this filter, whose purpose would be to distill a region like this down to a single signal of association, there remained 4 SNPs of similar effect in the post-filtered 237,602 SNPs. To prevent unwarranted inflation of the effect of this locus, the SNP from among the 4 with the highest call rate was selected to represent the locus, and the other 3 removed during the creation of the classification algorithms. This post-hoc procedure was only applied to the 4 9p21 SNPs, as 9p21 is clearly the most influential locus in the prediction of CAD.

Interestingly, this phenomenon appeared to be relatively constrained to 9p21, at least among the 'top SNPs'. As was mentioned earlier in the Discussion, Section 6.2.1, using the two different priors left approximately 25 and 100 SNPs with $\hat{P}(H_0^i) < 0.8$. Among those SNPs, which were the largest contributors to the algorithm, it did not appear that there were any SNPs other than at the 9p21 locus which had multiple SNPs from the same locus present; that is, all other loci were represented only once

in this set of top SNPs. Note that this could be a result of the fact that the filter generally worked quite well except for 9p21, or it could be caused by all the other top SNPs being artifacts and other nearby linked SNPs were truly not associated.

Therefore, it appears that while the filter was beneficial, further improvements are warranted, and some fine tuning would most likely help. This is especially true for 9p21, where SNPs had to be removed post-hoc, and for other reasons which have multiple moderately correlated SNPs ($r^2 < 0.8$ but $0.8 < D' < 1$) all being influenced by one true underlying signal. Although not discussed significantly in this thesis, using haplotypes may be beneficial in situations such as these, or other classification algorithms that are better able to classify using correlated data, such as Random Forests.

6.3.2 Posterior Probabilities

The original purpose of this thesis was not originally focused, or interested, with Bayesian Inference, but with classification algorithms in general. However, during the analysis of various classification algorithms, it was not felt that algorithms more complicated than the approach used by the Naive Bayes algorithm were warranted, or could be statistically validated. A more thorough discussion of classification algorithms analyzed during the course of this thesis follows in Section 6.4.

However, it was felt that the performance of the Naive Bayes algorithm may benefit from forcing the algorithm to consider only those SNPs which were most likely to be associated with disease; hence the 'modification' to the traditional Naive Bayes algorithm presented in the Methods chapter. This was done in an attempt to reduce the potential noise that would come from considering all SNPs, but to avoid an explicit feature selection phase, either by using a hard p-value cut off, which seemed arbitrary, or by using a dimension reduction algorithm, which hampers biological interpretation of results.

In an earlier version of this thesis, it was originally taken that $\hat{P}(H_0^i) = p$, where p is the p-value from a traditional statistical test taken with respect to the model being considered; Fisher's Exact Test for the Genotype model, and the Cochran Armitage Test for Trend for the Additive model. These p-values were subjected to either the Hochberg [23] or the False-Discovery Rate [5] correction for multiple testing, which was a similar approach to using the 'many' and 'few' priors in a Bayesian method. While this approach gave very similar results to those presented in the Application section, it was based upon the false assumption that a p-value will converge to 1 under the null hypothesis. The posterior probabilities used in the Bayesian method rectified this error and ensured statistical consistency of the modified Naive Bayes algorithm.

Had it been known beforehand that the simple estimation of $\hat{P}(H_0^i)$ by a p-value would not lead to a statistically consistent classification algorithm, but would require the extensive derivations shown in the Methods section, the approach may not have been used. Nonetheless, once the error was noticed in the p-value approach, significant attention was paid to ensure that the posterior probabilities were developed using a solid statistical framework. Furthermore, given the general concordance of rankings between p-value and posterior probability ranked SNPs, it was not felt that the calculation of $\hat{P}(H_0^i)$, and any assumptions used in its calculation, adversely affected the classification accuracy of the models.

6.3.3 Intra-Plate Similarity

One of the most apparent, and most unfortunate findings from the Application section was the discovery of an intra-plate bias among the genotypes. This means that genotypes from samples which were found on the same plate were more likely to be the same than samples plated on separate plates. This result was most apparent in the decreased cross-validation classification accuracy noticeable in Figure 5.5. This

has consequences not only for the classification algorithm work done in this thesis, but also draws into question the results of the traditional statistical analysis of this data which has already been performed.

There are several potential reasons which would account for an intra-plate bias, as was mentioned in Section 5.4, with none of them being particularly desirable. One of the most likely reasons is the calling algorithms which were employed, BRLMM and Birdseed, and the fact that these calling algorithms were generally run on one plate of samples at a time. For these two genotype calling algorithms, recall that genotyping is based on both prior and posterior estimates of cluster specific means and variances. The prior estimates of cluster means and variances should be relatively resilient to change with respect to the number of samples genotyped at once, as they are based on estimates across thousands of SNPs. However, the posterior estimates of cluster means and variances, which are influenced by only a single loci, may vary greatly depending on the samples genotyped.

For example, if one attempted to call genotypes for 100 subjects at a time, then for a SNP with a minor allele frequency of 10%, one would only expect only 1 homozygote minor allele sample per run on average for that SNP. Determining the mean and variance of this genotype 'cluster' is therefore difficult, and the algorithm may erroneously label this genotype as an outlier or as part of another cluster. Recall from earlier the discussion of the SNPs which had low posterior probabilities using the two priors; several of these had low minor allele frequencies, and upon closer inspection of these SNPs, instances were found in which the genotype distribution from one plate was substantially different from the rest, which is most likely attributable to incorrect estimation of posterior cluster means and variances at those SNPs (data not shown). It would be interesting to see how the results of this thesis would change if the genotype calling algorithms were rerun with a larger number of samples at once, say 1500 as opposed to 90, as this may attenuate the problem listed above. Furthermore, this may have a dramatic influence on the SNPs which have been identified in the

OHGS dataset through traditional statistical analysis as being most associated with CAD, the so called 'top-SNPs' of the OHGS study.

Another potential source of intra-plate bias could result from a genuine relatedness of samples which have been plated together, a form of bias which is not necessarily undesirable. Note that this is not just the bias that comes from plating cases and controls separately, which should not be affected by regular cross-validation versus plate specific cross-validation; but from plating subtypes of cases and controls together. For example, note that the CARA cases were all plated together on two plates, but are phenotypically more similar than the other cases, in that they all represent a more extreme form of the disease. Also, for the other two subgroups of cases, those collected through the lipid clinic of Dr. McPherson or through the catheterization lab following angiography, plating was normally done such that a plate contained only cases from one of the two subtypes. Note that the cases are slightly different phenotypically: the cases recruited through the McPherson lab were more varied, with some having not had either an angiogram or coronary revascularization but only myocardial infarctions, while those collected through the catherization lab all had at least one stenosis of at least 50%.

Additionally, the same may be said of the two subtypes of controls present in the OHGS, as there are a number of subtle differences between them and they were also usually plated by subtype. The two types of controls were the elderly asymptomatic healthy controls recruited by the McPherson lab and the angiographically clean individuals who came through the catherization lab. Despite the apparent similarities between the two groups of controls, there are still subtle differences between them, particularly the fact that the controls recruited by Dr. McPherson were all asymptomatic, even if a significant portion of them may still have coronary artery disease, while the controls recruited by the catherization lab were all free of coronary artery disease but were symptomatic enough to warrant an angiogram. Subtle differences in phenotype such as these may contribute to a *genuine* difference between plates, and

legitimate intra-plate similarity.

Lastly, it may be possible that some form of lab protocol has increased the apparent level of similarity of samples, be it contamination or some difference in the execution of the protocol between plates. Although it is possible, it would be very difficult to test to see what sort of an impact this had, save for actually rerunning some samples on different plates. Unfortunately, due to the high cost of genotyping, this is not a likely possibility.

Among these reasons, the genotype calling algorithm is the easiest to fix, and would only require re-running the algorithms to generate genotypes from the fluorescence intensity files. However, a short-term fix may also be an option. Recall the use of the HWE filter to detect likely sources of genotyping error. Although it was not shown, the distribution of the generated p-values was rather remarkable, in that there were a large number of samples with abyssmal p-values. In fact, the worst 1000 or so HWE p-values were smaller than 10^{-50}. It may be possible to introduce another filter to detect outlier plates within a single SNP, for each SNP, with the goal of either removing that plate or removing that SNP.

Consider the detection of an outlier plate within a SNP. One could perform, say, Fisher's Exact Test between the results generated by one plate versus the remaining plates, for each plate in turn, for each SNP on the array. To do this across all SNPs would require $n_{plates} \times n_{SNPs}$ p-value calculations, which, for our study with $n_{plates} = 39$ and $n_{SNPs} = 364,015$, would require about 14 million p-value calculations. If one considered this against a Bonferonni cutoff, the p-value threshold would be 3.5×10^{-9}, which does not seem unreasonable given the sort of p-values that genotyping errors can generate.

In conclusion, one of the sources of inter-plate variability, the way in which the genotype calling was performed, could be ameliorated rather easily. This could be accomplished by either attempting to minimize the most egregious calling algorithm errors using a new quality control filter, or by re-running the genotype calling algo-

rithms with as many samples as possible. It would also not be difficult to add one or two filters to detect SNPs that show differences between the case or control subtypes, much like the filter between the Affymetrix arrays, although it is unlikely this will remove very many SNPs. However, the laboratory genotyping errors, which cannot be avoided, are unfortunate realities which must be taken into consideration when analyzing a GWA study.

6.4 Other Classification Algorithms

In this section, we will discuss other classification algorithms which were considered and analyzed during the course of this thesis, and how they compare to the Naive Bayes classifier. These classifiers were predominantly decision tree style classifiers, which were considered due to their computational tractability at a high number of dimensions. Although these classifiers are certainly interesting, it was eventually decided that the Naive Bayes classification algorithm seemed like the most promising candidate algorithm for the purposes of this thesis.

This section will be organized as follows. First, decision trees as a form of 'base classifier' will be introduced, as will a method for constructing them, Classification And Regression Trees (CART). Following this, the second half of this subsection will deal with the Random Forest and Adaboost methods, which both use base classifiers such as CART. These two methods will be analyzed for their limiting behaviour and for their expected performance with respect to the classification task at hand.

Decision trees are classifiers which can be represented as acyclic directed tree graphs. Once a tree has been constructed by some method, classification is done by placing new samples at the root of the tree and then passing samples to subsequent nodes based on the splits at those nodes, until classification is ultimately done in a terminal node. The decision trees can be simple; for example, 'decision stumps' are classifiers in which classification is done with respect to a single variable. One could

consider a single-variable decision stump as a hyperplane classifier. More complicated decision trees can be thought of as a method of partitioning the sample space into nested hyperrectangles, where each hyperrectangle is free to take its own class.

To construct large decision trees, advanced methods are required, and one of the more popular of them is the Classification And Regression Trees (CART) algorithm, which was introduced in the early 1980's by Leo Breiman *et al.* [8]. The name refers to the fact that there are two different algorithms which produce two different types of output; classification trees when the objects are being classified to discrete classes and regression trees when the output is continuous. In this discussion, we will consider classification trees, as we are considering classification to a binary categorical output.

CART works by building a decision tree which has binary splits at its nodes, even if the predictor variables are continuous or multi-categorical. When building the tree, CART selects for each split the variable which increases the homogeneity of the subsequent nodes the most from among the set of all predictor variables, and as such tries to create terminal nodes which are as homogeneous as possible. Homogeneity in a node is usually measured with sort of entropy or impurity criterion. In CART, trees are usually not grown until termination, that is, the trees are not usually grown until all training samples fall into a homogeneous terminal node. Trees can either be be stopped according to prespecified terminal node formation rules, such as when a certain proportion of training samples in the node are of a given class, or according to other pruning rules, such as those which seek to set the number of terminal nodes at a prespecified number, such as 16 terminal nodes.

Several years after CART came out, Leo Breiman introduced a new classification algorithm in the journal *Machine Learning*, titled Random Forests (RF) [10]. Random Forests is a combining or ensemble classifier, in that it combines the results of many base classifiers in forming a final classifier. The version proposed in the *Machine Learning* paper was based on averaging the results of a large number of unpruned CART decision trees. Ultimate classification is done by a majority vote across all base

classifier decision trees, where each tree is given an equal weight. Note that variable selection in RF is different than in CART, in that at any node in a growing tree the best variable is selected not from the set of all predictor variables, but from a random subset of all variables. As such, the number of variables considered at any node is an important tuning parameter, which is usually tuned by running several instances of RF. Also, note that in RF, each tree is grown based on a different bootstrap sample of the original training data. Note that selecting a bootstrap sample of the training set is accomplished by selecting a random sample with replacement from the training set. Samples which are not selected in the bootstrap are said to be out-of-bag, and are used by RF to obtain internal cross-validation like estimates of classifier performance.

CART, along with other decision tree classifiers, are relatively easy to analyze. For example, one can usually prove statistical consistency by showing that two features hold true as the number of samples tend to infinity: the measure of the size of any partition representing a terminal node tends to 0; and the number of training sample subjects found in a terminal node tends to infinity. In fact, this thesis used a similar approach in proving consistency of the Naive Bayes estimator in Section 3.2.

However, Random Forests, unlike CART, is quite complex and random, and hence it is very difficult to analyze its limit behaviour. In one of his many Technical Reports, Breiman laments the lack of theoretical development in the field of combining classifiers as compared to the empirical work by titling one of the sections of his report "My Kingdom for Some Good Theoretical Explanations" [9]. More recent papers, including some by Breiman, have tried to resolve these issues. In an interesting paper discussing various consistency issues of Random Forests and averaging classifiers, Biau, Devroye and Lugosi (the same Devroye and Lugosi responsible for A Probabilistic Theory of Pattern Recognition [14]) prove a number of useful propositions [6]. For example, they show that an averaging classifier is consistent whenever the base classifier is, and that an averaging classifier based on bootstrapping is consistent whenever the base classifier is.

Unfortunately, the method in which CART can be shown to be consistent does not apply to RF, as RF has its decision trees grown until terminal nodes contain only a small number of samples. Further analysis is hampered by the fact that Random Forests greedily selects variables, that variable selection is done with respect to a random subset of all variables at a given node, and that Random Forests uses bootstrap training samples to grow its decision trees. All of these factors combine to make the analysis of the limit behaviour of RF rather difficult compared to that of the base classifiers, both in terms of consistency and in attempting to derive generalization error bounds.

That being said, one of the factors which made the Naive Bayes estimator proof work was the fact that variables were assumed to be independent. In all of the published works on RF, consistency proofs are usually not tackled with the benefit of such a strong assumption. It may be that if one assumes variables to be independent, then proving the consistency of Random Forests would be a straightforward procedure. Since no proof was readily available, and the benefits of Random Forests were not seen as being any greater than Naive Bayes if the variables are in fact independent, the analysis of this thesis focused on the Naive Bayes estimator. Furthermore, although never derived during the course of this thesis, probability bounds were thought to be much more easily derived for the Naive Bayes estimator than Random Forests.

Another method similar to Random Forests is the Adaboost method of Yoav Freund and Robert Schapire [18]. A simple yet essentially detailed description of the algorithm can be found in numerous publications, for example a conference paper by Reyzin and Schapire discussing classifier complexity [40]. Briefly, Adaboost is a weighted combining classifier, whereby each base classifier is constructed according to the same rules, but each successive classifier is built based on the performance of the classifiers which preceded it. Specifically, the weight given to each training sample member is initially set to be equal for all members, but weights are updated after the construction of a base classifier so that the samples misclassified by the newly made

base classifier are given more weight for the next iteration. Each of these rounds of base classifier construction are known as boosting, and allows each new classifier to be built with a focus for samples misclassified by its predecessors.

When compared to Random Forests, Adaboost is slightly simpler to analyze. In fact, Breiman showed in 2000 (although it may have been proven earlier) that Adaboost is consistent [9], by comparing it to a gradient descent approach through the set of linear combinations of all potential base classifiers. Note, however, that consistency when variables are not assumed to be independent requires that for base classifier decision trees, the number of terminal nodes exceeds the number of dimensions in the sample space.

Adaboost is also interesting in that a number of probability bounds have also been derived for its generalization error, such as one derived by Schapire *et al.* [43], which states that for a weighted classifier $f(x) = \text{sign}\left(\sum_i \alpha_i h_i(x)\right)$ with base classifiers $h_i(x)$,

$$P(f(x) \neq y) \leq \hat{P}(f(x) \neq y) + O\left(\sqrt{\frac{Td}{m}}\right)$$

where $P()$ refers to the real underlying probability, $\hat{P}()$ refers to the training set probability, d is the VC-dimension of the space of base classifiers, T is the number of boosting rounds (rounds of classifiers built), and m is the sample size of the training set. Note that the VC-dimension, short for the Vapnik-Chervonenkis dimension, is a measure of the size and flexibility of the set of all classifiers which can be generated from a particular classification algorithm (for the specific definition, please see A Probabilistic Theory of Pattern Recognition [14]).

However, despite the fact that this bound was not tight, Freund and Schapire noted that it did not reflect two important empirical observations. First, the performance of Adaboost usually improved as the number of rounds of boosting increased, for which the opposite might be predicted from the given bounds. Secondly, this increase usually continued after the training set error had dropped to zero. Based

on these observations, Schapire, Freund *et al.* derived a new bound based on the concept of the 'margin' [43]. If classification is assumed to be done to two classes, $\mathcal{Y} = \{-1, +1\}$, then the margin of a training sample with respect to a classifier $f(x)$ is $\text{margin}_f(x, y) = \frac{y\sum_i \alpha_i h_i(x)}{\sum_i \alpha_i}$, where one may think of $\sum_i \alpha_i h_i(x)$ as being akin to the Naive Bayes Score, *nbs*, and an attempt to approximate $\eta(x)$. Schapire and Freund go on to base their proof around a covering set approximating the space of all linear combinations of base classifiers, where using either the VC-dimension d or the cardinality of the set of all base classifiers, $|\mathcal{H}|$, they proved that

$$P(\text{margin}_f(x, y) \leq 0) \leq \hat{P}(\text{margin}_f(x, y) \leq \theta) + O\left(\sqrt{\frac{d}{m\theta^2}}\right) \tag{6.4.1}$$

$$P(\text{margin}_f(x, y) \leq 0) \leq \hat{P}(\text{margin}_f(x, y) \leq \theta) + O\left(\sqrt{\frac{\log|\mathcal{H}|}{m\theta^2}}\right) \tag{6.4.2}$$

Note that $\text{margin}_f(x, y) \leq 0 \iff f(x) \neq y$. Now, although these bounds seem impressive, the large number of SNPs in a GWA mean that these bounds will be quite large, as the number of classifiers considered and the VC-dimension of the space of classifiers will increase with the number of dimensions of the sample space. If we look at the actual form of the Theorem from [43] with regards to $\mathcal{H}$, then we see that

$$\begin{aligned} P(\text{margin}_f(x, y) \leq 0) \leq & \hat{P}(\text{margin}_f(x, y) \leq \theta) \\ & + 2\exp\left(-\frac{N\theta^2}{8}\right) + \sqrt{\frac{1}{2m}\log\left(\frac{N(N+1)^2|\mathcal{H}|^N}{\delta}\right)} \end{aligned} \tag{6.4.3}$$

where N relates to the size of the covering set for $\mathcal{H}$ in the proof, and where δ is such that the probability that the above holds is true with probability $1 - \delta$ with respect to the random selection of the training sample. Since N is a variable free to vary, the bound above can be optimized in terms of N. Indeed, the optimal bound as pointed

out by Freund in [18] is reached for

$$N = \frac{4}{\theta^2} \log\left(\frac{m}{\log|\mathcal{H}|}\right)$$

If we return to our interest, constructing a classifier from a genome wide association study, then consider the simplest possible set of base classifiers, where the base classifiers in $\mathcal{H}$ are decision stumps,

$$h_{i,j,k} = \begin{cases} (-1)^k, & x_i > j \\ (-1)^{k+1}, & x_i \leq j \end{cases} \tag{6.4.4}$$

then we have $|\mathcal{H}| = 237602 \times 2 \times 2$ and $m = 2997$, where m is the sample size. Note that $k \in \{0, 1\}$ flips classifier sign and $j \in \{0, 1\}$ allows the classifier to differentiate between 0 vs 12 and 01 vs 2.

Figure 6.1 gives an estimate of the error bound for various values of θ. Even for the simplest possible form of base classifiers considered, the error bounds are not good for $m = 2997$. Note that for interpretation of the above, this is not just an error bound on the training set error, but a bound on a type of training set error where samples which are classified with only marginal confidence for a given θ, those with $|\operatorname{margin}_f(x, y)| \leq \theta$ are deemed to be classified incorrectly. Preliminary analysis of the Adaboost classifier using decision stumps on the OHGS dataset did not yield an error bound on the generalization error of less than 1 for various values of θ (data not shown). As such, there were worries that the algorithm would significantly overfit the training set.

If we consider more complicated base classifiers than decision stumps, such as CART, then to consider generalization error bounds we must try to calculate some measure of the set of classifiers we are drawing from, such as the VC-dimension. However, it is very difficult to calculate the VC-dimension of CART. For example, in a discussion on performance bounds for the Adaboost method with CART base

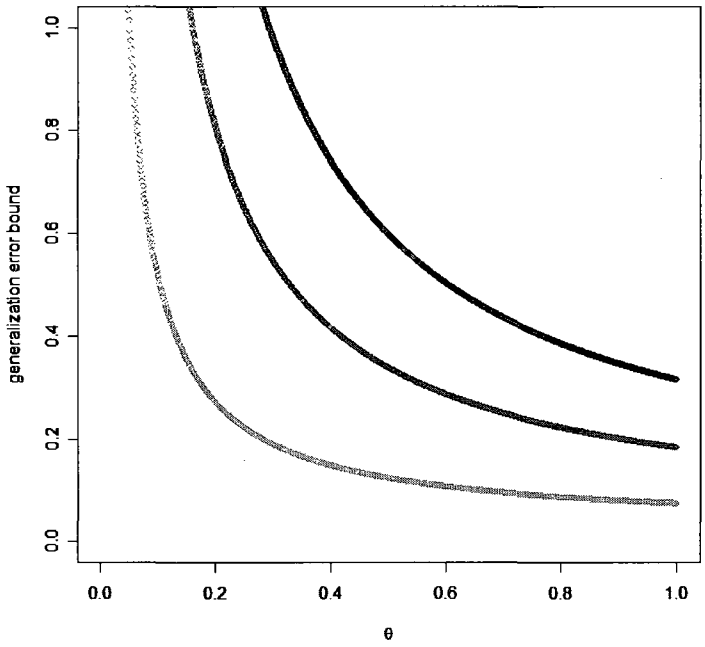

Figure 6.1: Generalization Error Estimates for Adaboost Method. Bounds are given for equation (6.4.3) with $|\mathcal{H}| = 950408$, $\delta = 0.001$, and $m = 2997$ (black/dark), $m = 10000$ (red/medium), and $m = 100000$ (green/light). Note that here m is the size of the training sample, not the number of dimensions, which is fixed.

classifiers, Reyzin and Schapire were forced to make their arguments with respect to the number of 'tree topologies' that CART generated for a certain number of terminal nodes [40]. So, while direct estimates of the generalization error for CART would be harder to come by than for decision stumps, they would necessarily be worse. It is hard to estimate whether the increase in classifier flexibility which results from using CART as opposed to decision stumps would make up for the increased potential for overfitting.

As such, even though Adaboost is consistent, given the number of training samples available and the generalization bounds which were provable, it was not felt that there was enough confidence in the eventual performance of the classifier to justify its use as opposed to the Naive Bayes classifier. As such, although eventual results

would no doubt have been interesting, for the purposes of this thesis, analysis was restricted to the Naive Bayes classifier.

6.5 Generalizability and Clinical Utility

This thesis is principally concerned with a method for predicting coronary artery disease using high dimensional genetic data from DNA microarrays. Although similar work does not yet exist in the literature given the relative infancy of GWAs, some groups have started to look at the benefit of individual SNPs in conjunction with typical risk factors to predict cardiovascular events. This section will be organized as follows. First, three papers will be discussed which use 9p21, the locus identified in the first wave of CAD/MI GWA publications, as a covariate in attempting to predict the long term risk of a cardiovascular event. Next, the results of these papers will be discussed with respect to the current state of CAD/MI GWAs, and what this means for attempting to quantify risk for subsequent cardiovascular loci with only moderate effect. Lastly, this section will consider the potential benefit of this thesis and case control studies in attempting to quantify the effect that genetics plays on a complex trait for use in predictive models.

An early paper published with respect to the effect of 9p21 on disease risk was by Talmud *et al.* in *Clinical Chemistry* in 2008 [46]. The study population in the paper was a prospective cohort study of healthy middle aged men which had 270 cardiovascular events among 2742 participants. In addition to the 9p21 genotype, age, triglycerides, cholesterol, smoking, systolic blood pressure and BMI were used as covariates for a Cox proportional hazards model. The authors note that 9p21 was significantly associated with outcome (hazard ratio 1.38 and 1.57 (p=0.04) for heterozygous and homozygous risk versus homozygous protection, respectively), and gave a better fit (LR p=0.01), but did not give a significant change in area under the ROC curve analysis. In attempting to rectify this, the authors perform the

statistically dubious procedure of adding nine other hypothetical SNPs, independent of 9p21 but with similar effect sizes and allele frequencies, and show that such a model would cause an area under the curve comparison to be statistically significant. In doing this procedure, the authors implicitly hint at the problem of proving the clinical utility of a single SNP, but that once additional variants are discovered, proving utility may be more straightforward. Unfortunately, in doing so the authors assumed future genetic variants found to be associated with CAD would be as powerful as 9p21; recent work, described in Section 6.1, has shown that future variants will be progressively smaller in their impact.

Following the Talmud paper, two more papers were published which used 9p21 genotype information to quantify long term risk, this time using longitudinal studies. These studies were published by Paynter *et al.* in *Annals of Internal Medicine* [35] and by Brautbar *et al.* in *Circulation Cardiovascular Genetics* [7]. The paper by Paynter, which had as the senior author Paul Ridker, was based on the Women's Genome Health Study (WGHS) of 22,129 Caucasian females, with 715 events. The paper by Brautbar, which has as senior author Eric Boerwinkle, was based on the Atherosclerosis Risk in Communities Study (ARIC) of 9,998 Caucasians, with 1349 events. Both papers clearly showed an association between 9p21 and risk: WGHS had hazard ratios of 1.25 and 1.32 ($p < 0.05$), adjusted for age, systolic blood pressure, triglycerides, smoking status, antihypertensive medication use, history of diabetes; ARIC had hazard ratios of 1.20 and 1.43 ($p = 3 \times 10^{-6}$), when adjusted for age, age-squared, gender, systolic blood pressure, triglycerides, diabetes status, smoking status and antihypertensive medication use. However, only ARIC calls these results significant with respect to the construction of the predictive model, based on a significant area under the curve comparison between models with and without 9p21. WGHS, despite the significance of the genotypes as variables in the model, does not report them as having had added significantly to the predictive value of the model.

The surprising aspect about the results from these three studies is the difficulty

they had in establishing 9p21 as a significant factor in predictive models for coronary artery disease. All three studies are of a relatively large size, although the number of events in the Talmud paper and WGHS are not as large as in the ARIC paper, which happened to be the only one to report that including 9p21 gave a significantly better model fit. However, it is worth noting that all three studies report that the 9p21 genotype variable on its own is significant in the model, which does speak to the benefit of including it in any model for cardiovascular disease. It is also interesting to note that in ARIC, the beta coefficient for 9p21 homozygous risk versus homozygous protection is 0.36, which is larger than antihypertensive medication use 0.1834 but less than current smoking status 0.4830 and diabetes 0.8349. Note that the beta coefficient for these studies is related to the variable's importance in predicting events, with a greater beta reflecting a more influential variable.

After reading these studies, one cannot help but think: What does this mean for other SNPs which have been shown to associate with CAD based on case-control studies? If we recall the timetable for the variants identified so far, Section 6.1, the Talmud, WGHS and ARIC papers were published after the first wave of GWAs, when only 9p21 was clearly associated with disease. It is important to note that 9p21 has a very large minor allele frequency and large odds ratio, and that SNPs identified in the second wave of GWAs have lower frequencies and lower odds ratios, giving 9p21 a larger effect in predictive models. It seems almost inevitable that if a second wave of longitudinal studies were published, they would have mixed results - that some of the 11 other loci associated with CAD would appear associated in one study and some in others. Given that these very large studies already have difficulty proving the predictive benefit of 9p21, the locus most clearly associated with CAD, it seems that there would be a lot of difficulty proving the predictive benefit of future variants, if those variants are included in a predictive model for CAD as individual variables.

To date, attempts to quantify the risk of multiple variants on a trait have been done rather simply. For example, papers attempting to quantify the overall risk

of multiple genetic variants for a disease, such as one by Lango *et al.* on diabetes in *Diabetes* [29], and by Kathiresan *et al.* on lipids in the *New England Journal of Medicine* [27] have been forced to use 'genotype risk scores' where each further copy of a risk allele at a locus increases a person's genotype risk score by 1, rather than multivariate models which attempt to quantify risk of alleles individually. This points to the potential utility of case control studies, and work such as done in this thesis, in establishing the relative contribution of various genetic factors in predicting genetic risk before moving to longitudinal type studies to demonstrate efficacy in predicting events. It remains an open question as to whether it would be better to construct these risk scores in prospective cohorts along with traditional risk factors, or to construct them in case-control studies and adjust for traditional risk factors in prospective studies. This may allow the genetic risk scores to be derived from complicated algorithms, while the ultimate model would be done in a conventional fashion such as logistic regression.

One of the benefits of the work done in this thesis is to explore methods for creating genotype risk scores separately from traditional risk factors, in populations of individuals which have good power and are genetically enriched for the trait of interest. Unfortunately for this thesis, the results were less than ideal, with sensitivities and specificities of the Naive Bayes estimator mirroring the effect of 9p21 almost exactly, and not seeming to draw from the additional wealth of information from the other SNPs. However, the course of this thesis did lead to numerous observations which will benefit future attempts at constructing classifiers from genome wide association data, and even on the traditional statistical analysis with regards to finding variants associated with disease. More complicated mathematical models such as those advanced in this thesis may play a better role in predicting disease than those which can be derived from traditional risk modeling strategies; however, it is likely that the best performance of those models would come from using a subset of clearly associated SNPs, and not the entire GWA dataset.

Chapter 7

Conclusion

The purpose of this thesis was to use techniques from machine learning to build a classification algorithm for coronary artery disease using genetic data from a genome wide association study. Although the classification algorithm was successfully constructed, the results of applying the algorithm to the genetic data from the OHGS were not as good as might be expected. The results of the Naive Bayes classifier, although promising when viewed through the results of cross-validation, were only as good as those of the single strongest locus, 9p21, on the test sets. However, this result is not an end in itself; during the course of this thesis, and the investigation of the constructed classification algorithm's unimpressive performance, several important realizations were made, and directions implied for future work.

One of the most important realizations was the intra-plate bias that was evident from the decrease in cross-validation accuracy when the OHGS training set was split randomly or by plate. This has ramifications not only for the construction of a classification algorithm, but for the interpretation of the results from the OHGS in a traditional fashion. Methods to accommodate this, by either introducing a new quality control filter or by re-running the genotype calling algorithms, could go a long way towards improving the confidence in some of the most highly differentiated

SNPs identified by the OHGS study.

With that in mind, and with the recent developments in the field of cardiovascular genome wide association studies and the promise of future studies on the horizon, one has to wonder whether the strategy adopted in this thesis is the most appropriate for constructing predictive algorithms. Keep in mind that this strategy was adopted when there was only one locus associated with coronary artery disease, and several probable associations of varying confidence. Now that there are a 10+ loci definitively associated with coronary artery disease, and plenty of other loci associated with related phenotypes such as lipid concentrations, blood pressure or diabetes, it may be possible to use more advanced algorithms on subsets of only a few dozen or hundred SNPs identified through genome wide association studies. Since these SNPs have been identified through numerous studies, they are all almost certainly real associations, and not statistical artifacts or genotyping anomalies. Fewer SNPs would also allow for the consideration of classification algorithms which were considered computationally unfeasible on the entire genome wide association dataset. As for the Naive Bayes classification algorithm itself, it may be possible to go beyond the restrictive independence assumption of (3.2.2), and create a Bayesian classifier which models a certain degree of interactions between loci.

In conclusion, although the Naive Bayes classifier worked, it did not work as well as anticipated. More accurate predictive models may come from constructing classification algorithms on proven sets of SNPs; from an even more rigorous quality control treatment of the SNPs under analysis; or from selecting classifiers from a more diverse set of classification algorithms.

Bibliography

[1] Affymetrix Inc (2006) BRLMM: an improved genotype calling method for the GenChip Human Mapping 500K Array Set. `http://www.affymetrix.com/support/technical/whitepapers/brlmm_whitepaper.pdf`

[2] Agresti A (1990) Categorial Data Analysis. John Wiley and Sons, Inc:New York.

[3] Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P (2002) Molecular Biology of the Cell, 4th Ed. Garland Science:New York.

[4] Armitage P (1955) Tests for Linear Trends in Proportions and Frequencies. Biometrics 11(3):375-386.

[5] Benjamini Y, Hochberg Y (1995) Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing. Journal of the Royal Statistical Society 57(1):289-300.

[6] Biau G, Devroye L, Lugosi G (2008) Consistency of Random Forests and Other Averaging Classifiers. Journal of Machine Learning Research 9:2015-2033.

[7] Brautbar A, Ballantyne CM, Lawson K, Nambi V, Chambless L, Folsom AR, Willerson JT, Boerwinkle E (2009) Impact of Adding a Single Allele in the 9p21 Locus to Traditional Risk Factors on Reclassification of Coronary Heart Disease Risk and Implications for Lipid-Modifying Therapy in the Atherosclerosis Risk in Communities Study. Circulation Cardiovascular Genetics 2:279-285.

[8] Breiman L, Friedman JH, Olshen RA, Stone CJ (1984) Classification and Regresstion Trees. 1st ed. Chapman & Hall/CRC:New York.

[9] Breiman L (2000) Some Infinity Theory for Predictor Ensembles. Technical Report 577, Statistics Department, University of California, Berkeley.

[10] Breiman L (2001) Random Forests. Machine Learning 45:5-32.

[11] Carlson CS, Eberle MA, Rieder MJ, Yi Q, Kruglyak L, Nickerson DA (2004) Selecting a Maximally Informative Set of Single-Nucleotide Polymorphisms for Association Analyses Using Linkage Disequilibrium. American Journal of Human Genetics 74:106-120.

[12] Casella G, Berger R (2002) Statistical Inference. 2nd ed. Duxbury:Canada.

[13] Coronary Artery Disease Consortium (2009) Large Scale Association Analysis of Novel Genetic Locis for Coronary Artery Disease. Arteriosclerosis, Thrombosis and Vascular Biology 29:774-780.

[14] Devroye L, Györfi L, Lugosi G (1996) A Probabilistic Theory of Pattern Recognition. Springer:New York.

[15] Di X, Matsuzaki H, Webster T, Hubbell E, Liu G, Don S, Bartell D, Huang J, Chiles R, Yang G, Shen M, Kulp D, Kennedy G, Mei R, Jones K and Cawley S (2005) Dynamic Model Based Algorithms for Screening and Genotyping Over 100K SNPs on Oligonucleotide Microarrays. Bioinformatics 21(9):1958:1963.

[16] Draghici S (2003) Data Analysis Tools for DNA Microarrays. Chapman & Hall/CRC:New York.

[17] Erdmann J, Großhennig A, Braund PS *et al.* (2009) New Susceptibility Locus for Coronary Artery Disease on Chromosome 3q22.3. Nature Genetics 41(3):280-282.

[18] Freund Y, Schapire RE (1997) A Decision-Theoretic Generatlization of On-Line Learning and an Application to Boosting. Journal of Computer and System Sciences 55:119-139.

[19] Harrell FE (2001) Regression Modeling Strategies. Springer-Verlag:New York.

[20] Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning. Springer:New York.

[21] Helgadottir A, Thorleifsson G, Manolescu A *et al.* (2007) A Common Variant on Chromosome 9p21 Affects the Risk of Myocardial Infarction. Science 316(8):1491-1493.

[22] Helgadottir A, Thorleifsson G, Magnusson KP *et al.* (2008) The Same Sequence Variant on 9p21 Associates with Myocardial Infarction, Abdominal Aortic Aneurysm and Intracranial Aneurysm. Nature Genetics 40(2):217-224.

[23] Hochberg Y (1988) A Sharper Bonferonni Procedure for Multiple Tests of Significance. Biometrika 75(4):800-802.

[24] Holm S (1979) A Simple Sequentially Rejective Multiple Test Procedure. Scandanavian Journal of Statistics 6:65-70.

[25] The International HapMap Consortium (2007) A Second Generation Human Haplotype Map of Over 3.1 Million SNPs. Nature 449:851-861.

[26] Johnson RA, Wichern DW (1982) Applied Multivariate Statistical Analysis. Prentice-Hall:New Jersey.

[27] Kathiresan S, Melander O, Anevski D, Guiducci C, Burtt NP, Roos C, Hirschhorn JN, Berglund G, Hedblad B, Groop L, Altshuler DM, Newton-Cheh C, Orho-Melander M (2008) Polymorphisms Associated with Cholesterol and Risk of Cardiovascular Events. New England Journal of Medicine 358:1240-1249.

[28] Korn JM, Kuruvilla FG, McCarroll SA, Wysoker A, Nemesh J, Cawley S, Hubbell E, Veitch J, Collins PJ, Darvishi K, Lee C, Nizzari MM, Gabriel SB, Purcell S, Daly MJ, Altshuler D (2008) Integrated genotype calling and association analysis of SNPs, Common Copy Number Polymorphisms and Rare CNVs. Nature Genetics 40(10):1253-1260.

[29] Lango H and the UK Type 2 Diabetes Genetics Consortium, Palmer CNA, Morris AD, Zeggini E, Hattersley AT, McCarthy MI, Frayling TM, Weedon MN (2008) Assessing the Combined Impact of 18 Common Genetic Variants of Modest Effect Sizes on Type 2 Diabetes Risk. Diabetes 57:3129-3135.

[30] Lucke JF (2008) A Critique of the False-Positive Report Probability. Genetic Epidemiology 33:145-150.

[31] McPherson R, Pertsemlidis A, Kavaslar N, Stewart AFR, Roberts R, Cox DR, Hinds DA, Pennacchio LA, Tybjaerg-Hansen A, Folsom AR, Boerwinkle E, Hobbs H, Cohen JC (2007) A Common Allele on Chromosome 9 Associated with Coronary Heart Disease. Science 316:1488-1491.

[32] Mitchell T (1997) Machine Learning. 1st ed. WCB/McGraw-Hill:New York.

[33] Myocardial Infarction Genetics Consortium (2008) Genome-Wide Association of Early-Onset Myocardial Infarction with Single Nucleotide Polymorphisms and Copy Number Variants. Nature Genetics 41(3):334-341.

[34] Patterson N, Price AL, Reich D (2006) Population Structure and Eigenanalysis. PLoS Genetics 2(12):2074-2093.

[35] Paynter NP, Chasman DI, Buring JE, Schiffman D, Cook NR, Ridker PM (2009) Cardiovascular Disease Risk Prediction With and Without Knowledge of Genetic Variation at Chromosome 9p21.3. Annals of Internal Medicine 150:65-72.

[36] Price AL, Pattern NJ, Plenge RM, Weinblatt ME, Shadick NA, Reich D (2006) Principal Components Analysis Corrects for Stratification in Genome-Wide Association Studies. Nature Genetics 38(8):904-909.

[37] Qiu W, Lazarus R (2009) Power Calculation for Testing If Disease is Associated with Marker in a Case-Control Study Using the GeneticsDesign Package. Part of supporting documentation for R GeneticsDesign package, `http://bioconductor.org/packages/2.4/bioc/manuals/GeneticsDesign/man/GeneticsDesign.pdf`

[38] R Development Core Team (2007) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, `http://www.R-project.org`

[39] Rabbee N, Speed T (2005) A Genotype Calling Algorithm for Affymetrix SNP Arrays. Bioinformatics 22(1):7-12.

[40] Reyzin L, Schapire RE (2006) How Boosting the Margin Can Also Boost Classifier Complexity. In Proceedings of the 23rd International Conference on Machine Learning. Accessed from `http://jmlr.csail.mit.edu/papers/volume8/bartlett07b/bartlett07b.pdf`

[41] Samani NJ, Erdmann J, Hall A, *et al.* (2007) Genomewide Association Analysis of Coronary Artery Disease. New England Journal of Medicine 357:443-453.

[42] SAS Institute Inc. 2004. SAS OnlineDoc 9.1.3. Cary, NC: SAS Institute Inc.

[43] Schapire RE, Freund Y, Bartlett P, Lee WS (1998) Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods. The Annals of Statistics 26(5):1651-1686.

[44] Slager SL, Schaid DJ (2001) Case-Control Studies of Genetic Markers: Power and Sample Size Approximations for Armitage's Test for Trend. Human Heredity 52:149-153.

[45] Stewart AFR, Dandona S, Chen L, Assogba O, Belanger M, Ewart G, LaRose R, Doelle H, Williams K, Wells GA, McPherson R, Roberts R (2009) Kinesin Family Member 6 Variant Trp719Arg Does Not Associate with Angiographically Defined Coronary Artery Disease in the Ottawa Heart Genomics Study. Journal of the American College of Cardiology 53(16):1471-1472.

[46] Talmud PJ, Cooper JA, Palmen J, Loverling R, Drenos F, Hingorani AD, Humphries SE (2008) Chromosome 9p21.3 Coronary Heart Disease Locus Genotype and Prospective Risk of CHD in Healthy Middle-Aged Men. Clinical Chemistry 54(3):467-474.

[47] Trégouët DA, König IR, Erdmann J *et al.* (2009) Genome-Wide Haplotype Association Study Identifies the SLC22A3-LPAL2-LPA Gene Cluster as a Risk Locus for Coronary Artery Disease. Nature Genetics 41(3):283-285.

[48] Wacholder S, Chanock S, Garcia-Closas M, El ghormli L, Rothman R (2004) Assessing the Probability That a Positive Report is False: An Approach for Molecular Epidemiology Studies. Journal of the National Cancer Institute 96(6)434:442.

[49] Warnes G, with Gorjanc G, Leisch F and Man M (2007) genetics: Population Genetics. R package version 1.3.2.

[50] Weir B (1996) Genetic Data Analysis II. Sinauer Associations, Inc:Sunderland, MA.

[51] Wellcome Trust Case Control Consortium (2007) Genome-Wide Association Study of 14,000 Cases of Seven Common Diseases and 3,000 Shared Controls. Nature 447(7):661-678.

Appendix A

Notation and Acronyms

This appendix contains a reference list of notation and acronyms which are commonly used throughout this thesis. Note that when used, hat notation usually refers to an estimate of an underlying probability or parameter, such as $\hat{P}(Y = 1|X = j)$ being an estimate for $P(Y = 1|X = j)$, the probability of being a case given case status.

Below is a list of notation based on the Greek alphabet.

$\eta(x)$	A probability measure giving the probability of being a case given a genotype; $\eta(x) = P(Y = 1\|X = x)$
$\mu(x)$	A probability measure giving the probability of a genotype; $\mu(x) = P(X = x)$
$\pi_{j\|i}$	Equivalent to $P(X = j\|Y = i)$
$\pi_{i\|j}$	Equivalent to $P(Y = i\|X = j)$

Below is a list of notation based on the Latin alphabet.

D'	Measure of LD between two SNPs. Ranges between 0 and 1, where $D' = 1$ indicates that no recombination is present between the two SNPs
g_n	A classifier built using a training set of n members; $g_n = g_n(x; s_n)$
g^*	Bayes Decision Rule, the best possible binary classification algorithm; $g^*(x) = \begin{cases} 1, & \eta(x) > \frac{1}{2} \\ 0, & \eta(x) \leq \frac{1}{2} \end{cases}$
$L(g)$	The generalization error of a classifier; $L(g) = P(g(X) \neq Y)$
L^*	The Bayes error; $L^* = L(g^*)$ for g^* the Bayes Decision Rule
L_n	Conditional probability of error for a classification algorithm; $L_n = L(g_n) = P(g_n(X; S_n) \neq Y \mid S_n)$
m or M	Depends on context; often the number of SNPs / the number of dimensions of sample space, $\dim(\mathcal{X}) = m$
n or N	Depends on context; usually the size of the training sample
$n_{i,j}$	The number of people of class i with genotype j, $i \in \{0, 1\}$, $j \in \{0, 1, 2\}$
$n_{i,+}$	The number of people of class i
$n_{+,j}$	The number of people with genotype j
n_A	The number of A alleles found in the training sample for a SNP
n_{AA}	The number of people with genotype AA for a SNP
n_{AABB}	The number of people with genotype AA at one SNP with alleles A and a, and genotype BB at a second SNP with alleles B and b
n_{ab}^{AB}	The number of people who have haplotype AB on one chromosome and haplotype ab on the other chromosome

$nbs(x)$	Naive Bayes Score for a person with genotype x, so that $nbs(x) : \mathcal{X} \to \mathbb{R}$; $nbs(x)$ allows for a classifier $g(x)$ to be built with threshold θ such that $g(x;\theta) = \begin{cases} 1, & nbs(x) \geq \theta \\ 0, & nbs(x) < \theta \end{cases}$
p_A	The allele frequency of allele A
p_{AB}	The haplotype frequency of haplotype AB
p_{AABB}	The proportion of people with genotype AA at one SNP with alleles A and a, and genotype BB at a second SNP with alleles B and b
p_{ab}^{AB}	The proportion of people who have haplotype AB on one chromosome and haplotype ab on the other chromosome
r^2	Measure of LD between two SNPs. Ranges between 0 and 1, where $r^2 = 1$ indicates a one-to-one relationship between genotypes at the two SNPs
s_n	A particular realization of a training set with n members; $s_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$
S_n	The random variable version of a training set; $S_n = ((X_1, Y_1), \ldots, (X_n, Y_n))$ drawn from $(\mathcal{X} \times \mathcal{Y})^n$ with respect to some probability measure
x	Depends on context; often an m dimensional vector of genotypes, $x \in \mathcal{X}$
$\mathcal{X}$	The sample space of genotypes; $\mathcal{X} = \{0, 1, 2\}^m$
y	Depends on context; often class membership/case-control status of a person, $y \in \mathcal{Y}$
$\mathcal{Y}$	Space of class membership; $\mathcal{Y} = \{0, 1\}$, where 0 is control and 1 is case

Below is a list of acronyms which are commonly used in this thesis

ARIC	Atherosclerosis Risk in Communities Study
BRLMM	Bayesian Robust Linear Model with Mahalanobis distance classifier
CAD	Coronary Artery Disease
CARA	Coronary Artery bypass Risk Assessment
CHD	Coronary Heart Disease
CART	Classification And Regression Trees
CCF	Cleveland Clinic Foundation
CR	Call Rate
CV	Cross-Validation
DNA	DeoxyriboNucleic Acid
DM	Dynamic Module
FDR	False Discovery Rate
FN	False Negative
FP	False Positive
FWER	Family Wise Error Rate
GWA	Genome Wide Association
HWE	Hardy-Weinberg Equilibrium
LD	Linkage Disequilibrium
MAF	Minor Allele Frequency
MI	Myocardial Infarction
NBS	Naive Bayes Score or National Blood Services
OHGS	Ottawa Heart Genomics Study
PCA	Principal Components Analysis
SNP	Single Nucleotide Polymorphism
RF	Random Forests
ROC	Receiver Operator Characteristic

FP	False Positive
TP	True Positive
UOHI	University of Ottawa Heart Institute
VC	Vapnik-Chervonenkis
WHGS	Women's Genome Health Study
WTCCC	Wellcome Trust Case Control Consortium

Appendix B

Code

B.1 Common Variables

The following lists the code of `0_common_declarations.r` from Figure 4.1. It is a simple program designed to create a file with variables common to all other programs.

The code for `0_common_declarations.R`

```
#comment in path in which central 'thesis_work' folder is located
#path='c:/Documents and Settings/OHIUser/Desktop/Robbie/'
path='c:/R/'
test_master=c('genoexactp','additivep') #Genotype or Additive probability models
correct_master=c('many','few','none') #new selection by prior probability
trial_master=c('WTCCC','WTCCC','OHGS','CCF') #Cohort names
trial_submaster=c('WTCCC_case','WTCCC_ctrl','OHGS','CCF') #Cohort subnames - WTCCC is split up as it is too large to fit into RAM in one piece
group_length=c(1926,2938,2997,1920) #number of samples in each cohort
case_or_control=integer(2997) #case (1) or control (0) for OHGS
case_or_control[1:1542]=as.integer(1)
max_risk=0.8 #in Naive Bayes, setting 1-0.8=0.2<p(y=1|x)<0.8
r_squared_threshold=0.8 # for the r2 filter

K=as.single(0.10)
```

```
p_H0=as.single(c(1-1/2000,1-1/20000))
RR_high_max=as.single(2)
p_H0_t=as.single(integer(length(p_H0)))



save(list=ls(all=TRUE),file=paste(path,'thesis_work/0_common_
    variables/common_variables.Rdata',sep=""))
```

B.2 r^2 Filter

The following lists the code of `4_OHGS_r_squared_filter.r` from Figure 4.1. It runs the r^2 filter described in Section 3.1.

Data enters the r^2 filter separately by chromosome, and has not yet been filtered. Two important matrices to consider are `snp_data`, an $n_{SNP\ on\ chromosome\ i}$ by $n_{subjects}$ matrix where `snp_data[i,j]` gives the genotype of subject j for SNP i, and `snp_data_positions_large`, an n_{SNP} by about 20 matrix containing information such as SNPid, rsid, physical position and quality control information for each of the SNPs on the array.

The code for `4_ohgs_r_squared_filter.R`

```
#path='C:/R/'
path='c:/Documents and Settings/OHIUser/Desktop/Robbie/'

load(file=paste(path,'thesis_work/0_common_variables/common_
    variables.Rdata',sep=''))
load(file=paste(path,'thesis_work/4_filtered/OHGS_QC_snp_list.
    Rdata',sep=''))

#set current trial name - or can run loop wrt master_list
trial_name='OHGS'

snp_data_r2_filter=cbind(list_of_QC_SNPs,'A',0) # nSNP x 3 matrix
     with column 1 being the SNPid, column 2 being the replacement
     SNPid (or itself), and column 3 being a 1 (include) or 0 (
    remove) variables
```

```
for(chromosome_number in 22:1)
{
  load(file=paste(path,'thesis_work/3_unfiltered_with_QC_
      information/ohgs_chromosome_number_',chromosome_number,'_QC.
      Rdata',sep=''))

  #apply filter to chromosome data
  colnames(snp_data_positions)[colnames(snp_data_positions)=='
      SNPid']='snpid'
  snp_data_positions2=cbind(snp_data_positions,1:dim(snp_data_
      positions)[1])
  colnames(snp_data_positions2)[dim(snp_data_positions2)[2]]='pos
      '
  list_of_QC_SNPs2=cbind(list_of_QC_SNPs,0)
  colnames(list_of_QC_SNPs2)[1]='snpid'
  snp_data_positions3=merge(snp_data_positions2[,c('snpid','pos')
      ],list_of_QC_SNPs2,all.x=T,by='snpid')
  snp_data_positions3=snp_data_positions3[is.na(snp_data_
      positions3[,3])==F,]
  keep_snps_list=sort(snp_data_positions3[,'pos'])

  ##now have keep_snps_list - a list of snps to keep, in order,
      for this chromosome
  snp_data_positions=snp_data_positions[keep_snps_list,]
  snp_data=snp_data[keep_snps_list,]
  snp_data_controls=snp_data[,case_or_control==0]

  ## start filtering process
  # matrix of central importance is block1, which for each 'block
      ', gives number of SNPs is block in column 1, then in column
       2 through n+1 gives identifier for n SNPs in block
  num_of_snps=dim(snp_data)[1]
  un_samp_snps=1:num_of_snps
  len_un_samp_snps=length(un_samp_snps)
  block1=matrix(c(1:(num_of_snps*50)-1:(num_of_snps*50)),nrow=num
      _of_snps)
  if(chromosome_number==17)
    block1=matrix(c(1:(num_of_snps*200)-1:(num_of_snps*200)),nrow
        =num_of_snps)
  snp_bp_master=as.integer(as.character(snp_data_positions[,'
      PhysPosition']))
  run_num=0
  snp_MAF_master=as.numeric(as.character(snp_data_positions[,'
```

```
    study_MAF'])) #MAF is important because only SNPs within a
    certain MAF can have an r^2 over a certain threshold
snp_MAF_master=snp_MAF_master/100
missing_master=as.integer(as.character(snp_data_positions[,'
    ctrlN']))
missing_master=(dim(snp_data_controls)[2]-as.integer(as.
    character(snp_data_positions[,'ctrlN'])))/dim(snp_data_
    controls)[2]

while(len_un_samp_snps>0)
{
  run_num=run_num+1;
  cur_snp=un_samp_snps[1];
  check_snp_pos=2
  check_snp=un_samp_snps[check_snp_pos];
  block1[run_num,1]=1
  block1[run_num,2]=cur_snp
  cur_snp_bp=snp_bp_master[cur_snp]

  pA=snp_MAF_master[cur_snp]
  upper_bound=pA/(pA+r_squared_threshold*(1-pA))
  lower_bound=r_squared_threshold*pA/((1-pA)+r_squared_
      threshold*pA)

  pA_2=snp_MAF_master[cur_snp]+missing_master[cur_snp]+missing_
      master[check_snp]
  upper_bound_2=pA_2/(pA_2+r_squared_threshold*(1-pA_2))+
      missing_master[cur_snp]+missing_master[check_snp]
  pA_2=snp_MAF_master[cur_snp]-missing_master[cur_snp]-missing_
      master[check_snp]
  lower_bound_2=r_squared_threshold*pA_2/((1-pA_2)+r_squared_
      threshold*pA_2)-missing_master[cur_snp]-missing_master[
      check_snp]
  #one can calculate the upper and lower bound on MAF, beyond
      which r2<r2_threshold
  #speeds up calculations by at least an order of magnitude

  while(abs(cur_snp_bp-snp_bp_master[check_snp])<1000000 & (is.
      na(check_snp)==F)) #if SNP is within window
  {

    if(lower_bound_2<snp_MAF_master[check_snp] & snp_MAF_master
        [check_snp]<upper_bound_2) #if SNP is within MAF window
    {
```

```
        r_cur=r_unsquared(snp_data_controls[cur_snp,],snp_data_
            controls[check_snp,]) #current r_squared
        if(r_cur==-3) #indicates problem with r2 calculation -
            was necessary during development of r2 calculator, to
            deal with unforseen special cases in genotype
            distributions
74        print(c(cur_snp,block1[run_num,i]+gotten_rid_of))

        if(r_cur*r_cur>r_squared_threshold) #yay! may be able to
            add SNP to block
        {
          if(block1[run_num,1]>1) #meaning there is more than one
              SNP in the block
79        {
            tvar=0;
            for(i in 3:(block1[run_num,1]+1)) #check it against
                all the SNPs in the block
            {
              r_cur=r_unsquared(snp_data_controls[check_snp,],snp
                  _data_controls[block1[run_num,i],])
84            if(r_cur*r_cur<r_squared_threshold) #this means it
                  fails the block
                tvar=1
            }
            if(tvar==0) #good - add to the block
            {
89            block1[run_num,block1[run_num,1]+2]=check_snp
              block1[run_num,1]=block1[run_num,1]+1;
              un_samp_snps=un_samp_snps[(un_samp_snps==check_snp)
                  ==F]
              len_un_samp_snps=len_un_samp_snps-1
              check_snp=check_snp-1
94          }
          } #end of if more than one SNP in block

          if(block1[run_num,1]==1) #only 1 SNP in block - ie this
              is the first snp to be added to this block
          {
99          block1[run_num,1]=2
            block1[run_num,3]=check_snp
            un_samp_snps=un_samp_snps[(un_samp_snps==check_snp)==
                F]
            len_un_samp_snps=len_un_samp_snps-1
            check_snp=check_snp-1
```

```
104           } #end of only 1 SNP of block

            } # end of r*r>r2_threshold
          } #end of MAF check

109       #can now move on to next SNP in window
          check_snp_pos=check_snp_pos+1
          check_snp=un_samp_snps[check_snp_pos]

        } # end of check snps in the window
114
        #window exhausted - move to the next snp down the line
        un_samp_snps=un_samp_snps[(un_samp_snps==cur_snp)==F]
        len_un_samp_snps=len_un_samp_snps-1

119     if((run_num%%100)==0) #periodically print updates to make
          sure program is running smoothly
        print(c(chromosome_number,len_un_samp_snps,num_of_snps))

      } #end of chromosome for r^2 filter

124   block1=block1[1:run_num,1:(1+max(block1[,1]))] #truncate block1
        to fit results

      #Done step 1 - generating 'block1', a matrix of blocks with how
        many snps in a block and which snps they are. now need to
        select representative snp from each block. currently taking
        best call rate

      snp_data_include_list=matrix(rep(c('','',0),dim(snp_data_
        controls)[1]),ncol=dim(snp_data_controls)[1])
129   snp_data_include_list=t(snp_data_include_list)
      snp_data_include_list[,1]=as.character(snp_data_positions[,'
        snpid'])

      for(i_snp in 1:dim(block1)[1]) #selection process to take best
        SNP in each block
      {
134     cur_snps=block1[i_snp,2:(block1[i_snp,1]+1)]
        keep_snp=cur_snps[which.min(missing_master[block1[i_snp,2:(
          block1[i,1]+1)]])]
        snp_data_include_list[cur_snps,2]=snp_data_include_list[keep_
          snp,1]
        snp_data_include_list[cur_snps,3]=0
```

```
    snp_data_include_list[keep_snp,3]=1
  } #end of i_snp counter

  #now add results to snp_data_r2_filter
  snp_match=match(snp_data_include_list[,1],snp_data_r2_filter
      [,1])
  snp_data_r2_filter[snp_match,]=snp_data_include_list

  save(snp_data_r2_filter,file=paste(path,'thesis_work/4_filtered
      /ohgs_r2_',r_squared_threshold,'_filter.Rdata',sep=""))

} #end of chromosome_number


#
## Functions used in the above
#


#####Define r_unsquared function
r_unsquared=function(data1,data2)
{
  x=table(as.integer(data1)*3+as.integer(data2))
  ct=matrix(c(0,0,0,0,0,0,0,0,0),nrow=3)
  ct[1,1]=as.numeric(x['0'])
  ct[1,2]=as.numeric(x['3'])
  ct[1,3]=as.numeric(x['6'])
  ct[2,1]=as.numeric(x['1'])
  ct[2,2]=as.numeric(x['4'])
  ct[2,3]=as.numeric(x['7'])
  ct[3,1]=as.numeric(x['2'])
  ct[3,2]=as.numeric(x['5'])
  ct[3,3]=as.numeric(x['8'])

  if(is.na(x['0']))  ct[1,1]=0
  if(is.na(x['3']))  ct[1,2]=0
  if(is.na(x['6']))  ct[1,3]=0
  if(is.na(x['1']))  ct[2,1]=0
  if(is.na(x['4']))  ct[2,2]=0
  if(is.na(x['7']))  ct[2,3]=0
  if(is.na(x['2']))  ct[3,1]=0
  if(is.na(x['5']))  ct[3,2]=0
  if(is.na(x['8']))  ct[3,3]=0

```

```
n=sum(ct);
pAB_AB=ct[1,1]/n;
pAB_Ab=ct[1,2]/n;
pAB_aB=ct[2,1]/n;
p_flat_AaBb=ct[2,2]/n;
pA=ct[1,1]+ct[1,2]+ct[1,3]+0.5*(ct[2,1]+ct[2,2]+ct[2,3])
pA=pA/n
pB=ct[1,1]+ct[2,1]+ct[3,1]+0.5*(ct[1,2]+ct[2,2]+ct[3,2])
pB=pB/n

M=-2*pAB_AB-pAB_Ab-pAB_aB;

a=4;
b=2*(1-2*pA-2*pB)+2*M-p_flat_AaBb;
c=2*(pA*pB)+M*(1-2*pA-2*pB)-p_flat_AaBb*(1-pA-pB);
d=M*(pA*pB);

z=c(d,c,b,a);
r=polyroot(z);
realvar=c(0,0,0);greatestvar=0;out2=0;

for(i in 1:3)
{
  if(abs(Im(r[i]))<1e-8)
  {
    t2=Re(r[i])
    t=t2
    t3=0
    if(abs(pA-t2)<1e-10)
    {
      t=pA
      t3=1
    }
    if(abs(pB-t2)<1e-10)
    {
      t=pB
      t3=1
    }
    if(abs(1-pA-pB+t2)<1e-10)
    {
      t=pA+pB-1
      t3=1
    }
    if(abs(t)<1e-10)
```

```
224       {
            t=0
            t3=1
          }
          if(t3==1)
229       {
            if(greatestvar==0)
            {
              greatestvar=i
              realvar[i]=-1e8
234         }
            if(greatestvar>0)
            {
              if((((Re(r[i])-pA*pB)/(pA*(1-pA)*pB*(1-pB))^0.5)^2)>((
                  Re(r[greatestvar])-pA*pB)/(pA*(1-pA)*pB*(1-pB))^0.5)
                  ^2)
              {
239             greatestvar=i
                realvar[i]=-1e8
              }
            }
          }
244       if(t<0)
            t3=1
          if(t3==0 & ((pA-t)>0) & ((pB-t)>0) & ((1-pA-pB+t)>0) ) #
              good to go
          {
            realvar[i]=(2*ct[1,1]+ct[1,2]+ct[2,1])*log(t)+(2*ct[1,3]+
                ct[1,2]+ct[2,3])*log(pA-t)+(2*ct[3,1]+ct[2,1]+ct[3,2])
                *log(pB-t)+(2*ct[3,3]+ct[2,3]+ct[3,2])*log(1-pA-pB+t)+
                ct[2,2]*log(t*(1-pA-pB+t)+(pA-t)*(pB-t))
249         if(realvar[i]=='NaN')
              out2=1
            if(greatestvar<1)
              greatestvar=i;
            if(greatestvar>1)
254         {
              if(Re(realvar[i])>Re(realvar[greatestvar]))
              {
                greatestvar=i;
              }
259         }
          } #end of if #good to go
        } #end of if not imaginary
```

```
  } #end for i in 1:3

264 if(greatestvar>0)
  {
    t=Re(r[greatestvar])
    r_notsquared=(Re(r[greatestvar])-pA*pB)/(pA*(1-pA)*pB*(1-pB))
        ^0.5;
  }
269 if(greatestvar==0)
    r_notsquared=-2

  if(out2==1)
    out=-3
274 if(out2==0)
    out=r_notsquared
}
```

B.3 Naive Bayes

The following lists the code of `6_A_Naive_Bayes.r`, `6_B_Naive_Bayes_CV.r` and `6_C_Naive_Bayes_Main_Function.r` from Figure 4.1. Construction of the Naive Bayes classifier is done in `6_C`, trained on either the entire dataset `6_A` or some form of Cross-Validation `6_B`.

The code for `6_A_Naive_Bayes.r`

```
path='c:/Documents and Settings/OHIUser/Desktop/Robbie/'
#path='C:/R/'
load(file=paste(path,'thesis_work/0_common_variables/common_
    variables.Rdata',sep=''))
4
training_name='OHGS' #for alternative training/testing versions

### load or unload fortran programs
dyn.unload(paste(path,'thesis_work/fortran_programs/prior_scripts
    _4.dll',sep=""))
9 dyn.load(paste(path,'thesis_work/fortran_programs/prior_scripts_
    4.dll',sep=""))
##################################
```

```
### load summary data
load(file=paste(path,'thesis_work/4_filtered/',training_name,'_QC
    _and_r2_',r_squared_threshold,'_summary_data.Rdata',sep='')) #
    load summary data
###add extra columns
snp_data_positions_large=cbind(snp_data_positions_large
    ,1,1,0,1,1,0)
colnames(snp_data_positions_large)[(dim(snp_data_positions_large)
    [2]-5):dim(snp_data_positions_large)[2]]=
  c('p_H0_t_additivep_many','p_H0_t_additivep_few','p_H0_t_
     additivep_none',
    'p_H0_t_genoexactp_many','p_H0_t_genoexactp_few','p_H0_t_
     genoexactp_none')

### Need to add p_H0_t for two models for all SNPs
snp_data_temp_case=cbind(as.integer(snp_data_positions_large[,'
    caseAA'])
    ,as.integer(snp_data_positions_large[,'caseAB'])
    ,as.integer(snp_data_positions_large[,'caseBB']))
snp_data_temp_ctrl=cbind(as.integer(snp_data_positions_large[,'
    ctrlAA'])
    ,as.integer(snp_data_positions_large[,'ctrlAB'])
    ,as.integer(snp_data_positions_large[,'ctrlBB']))
snp_data_temp_pvalues=array(1,c(dim(snp_data_positions_large)
    [1],6))

K=as.single(0.10)
p_H0=as.single(c(1-1/2000,1-1/20000))
RR_high_max=as.single(2)
p_H0_t=as.single(integer(length(p_H0)))

for(i in 1:dim(snp_data_positions_large)[1])
{
     t1=as.single(snp_data_temp_case[i,])
     t2=as.single(snp_data_temp_ctrl[i,])
     if(sum(t1)+sum(t2)>0)
     {
       n_it=as.integer(1000)
       snp_data_temp_pvalues[i,1:2]=as.numeric(unlist(.Fortran("
          trend_prior_",t1,t2,K,p_H0,n_it,RR_high_max,p_H0_t,as.
          integer(length(p_H0)))[7]))
       n_it=as.integer(100)
       snp_data_temp_pvalues[i,3:6]=as.numeric(unlist(.Fortran("
```

```
          genotype_prior_",t1,t2,K,p_H0,n_it,RR_high_max,p_H0_t,as
          .integer(length(p_H0)),p_H0_t)[c(7,9)]))
    }
     if((i%%10000)==0)
       print(c(as.character(i),as.character(dim(snp_data_positions
          _large)[1]),date()))
  }
}

###add data in
snp_data_positions_large[,'p_H0_t_additivep_many']=    as.numeric(
    snp_data_temp_pvalues[,1])
snp_data_positions_large[,'p_H0_t_additivep_few']=     as.numeric(
    snp_data_temp_pvalues[,2])
#snp_data_positions_large[,'p_H0_t_genoexactp_many_old']=    as.
    numeric(snp_data_temp_pvalues[,3])
#snp_data_positions_large[,'p_H0_t_genoexactp_few_old']=     as.
    numeric(snp_data_temp_pvalues[,4])
snp_data_positions_large[,'p_H0_t_genoexactp_many']=    as.numeric
    (snp_data_temp_pvalues[,5])
snp_data_positions_large[,'p_H0_t_genoexactp_few']=     as.numeric
    (snp_data_temp_pvalues[,6])



####DANGEROUS - initiate group_risk - matrix which stores Naive
    Bayes Scores
group_risk=array(0,c(length(correct_master),length(test_master),
    length(trial_submaster),max(group_length)))
####

### Time for Naive Bayes
for(i_test in 1:2)
{
  for(i_correct in 1:3)
  {
    test_set=array(T,length(case_or_control))
    #training_snps
    training_set_available=test_set_concordant
    training_set_available[c(130233,130235,130236)]=F

    group_risk=naive_bayes_main_function(test_set,i_test,i_
        correct,group_risk)
    save(group_length,group_risk,file=paste(path,'thesis_work/6_A
```

```
        _Naive_Bayes/naive_bayes_r2_',r_squared_threshold,'_
        regular_',training_name,'_training_set.Rdata',sep=""))
  } #end i_correct
} #end i_test
```

The code for 6_B_Naive_Bayes_CV.r

```
path='c:/Documents and Settings/OHIUser/Desktop/Robbie/'
#path='C:/R/'
load(file=paste(path,'thesis_work/0_common_variables/common_
    variables.Rdata',sep=''))

training_name='OHGS' #for alternative training/testing versions

### load or unload fortran programs
#dyn.unload(paste(path,'thesis_work/fortran_programs/hochberg_fdr
    .dll',sep=""))
#dyn.load(paste(path,'thesis_work/fortran_programs/hochberg_fdr.
    dll',sep=""))
dyn.unload(paste(path,'thesis_work/fortran_programs/prior_scripts
    _4.dll',sep=""))
dyn.load(paste(path,'thesis_work/fortran_programs/prior_scripts_
    4.dll',sep=""))
###

load(file=paste(path,'thesis_work/4_filtered/',training_name,'_QC
    _and_r2_',r_squared_threshold,'_summary_data.Rdata',sep='')) #
    load summary data

number_of_snps=dim(snp_data_positions_large)[1]
#temp introduction of case_or_control - OHGS style
case_or_control=integer(2997)
case_or_control[1:1542]=1
case_or_control=as.integer(case_or_control)

#need to select cross validation scheme
load(file=paste(path,'thesis_work/6_A_Naive_Bayes/plate_number.
    Rdata',sep=''))
sample_list=read.table(file=paste(path,'thesis_work/6_A_Naive_
    Bayes/cad1542ctrl1455samplelist.txt',sep=''),header=T)
#need to make the names the same
sample_list_short=array('',c(dim(sample_list)[1],1))
colnames(sample_list_short)=c('patientID_STD')
for(i in 1:dim(sample_list)[1])
```

```
{
  t1=unlist(strsplit(as.character(sample_list[i,1]),split=''))
  t1a=t1=='-'
  t1=t1[F==t1a]
  t2=as.integer(t1)
  sample_list_short[i]=as.character(paste(c(t1[is.na(t2)],as.
      integer(paste(t1[F==is.na(t2)],collapse=''))),collapse=''))
}
#need to merge plate info from plate_number into temp2
sample_list_short=cbind(sample_list_short,1:length(sample_list_
    short))
colnames(sample_list_short)=c('patientID_STD','pos')
x=merge(sample_list_short,plate_number,by='patientID_STD',all.x=T
    ,sort=FALSE)
x=x[order(as.integer(as.character(x[,'pos']))),]
plate=as.character(x[,'plate'])
plate[is.na(plate)]='unknown'

#now that the plate number is known, can run CV
subject_CV=array(length(case_or_control))
CV_folds=length(table(plate))
#mess up the order on purpose
fold_mess=sample(CV_folds)
for(i_fold_count in 1:(CV_folds))
{
  i_fold=fold_mess[i_fold_count]
  subject_CV[plate==names(table(plate)[i_fold_count])]=i_fold
}
table(paste(subject_CV,plate))
version_master=c('by_plate','by_plate_random')

if(1==0) #randomize CV fold assignment
{
  CV_folds=10
  subject_CV=sample(10,2997,replace=T)
  version_master=c('random')
}


for(i_block in 1:CV_folds)  #Can now do calculations for each
    block/fold in CV scheme
{

  for(i_version in 1:length(version_master)) #Either randomized
```

```
    or by plate
{
  if(i_block>1)
    load(file=paste(path,'thesis_work/6_A_Naive_Bayes/naive_
        bayes_r2_',r_squared_threshold,'_Cross_validation_',
        version_master[i_version],'_',CV_folds,'_folds_by_plate.
        Rdata',sep=""))

###DANGEROUS - initiate group_risk
if(i_block==1)
  group_risk=array(0,c(length(correct_master),length(test_
      master),length(trial_submaster),max(group_length)))

###DANGEROUS - perform randomization
if(i_block==1 & i_version==2)
  subject_CV=subject_CV[sample(length(case_or_control))]

remove(merged_data)
gc(reset=T)
test_set=subject_CV==i_block
train_set=subject_CV!=i_block
train_set_cases=train_set&(case_or_control==1)
train_set_controls=train_set&(case_or_control==0)
load(file=paste(path,'thesis_work/5_merged_data/OHGS_merged_
    QC_r2_',r_squared_threshold,'_data.Rdata',sep=""))

remove(snp_data_positions_large)
#load summary data for another pass
load(file=paste(path,'thesis_work/4_filtered/',training_name,
    '_QC_and_r2_',r_squared_threshold,'_summary_data.Rdata',
    sep=''))

#add columns
snp_data_positions_large=cbind(snp_data_positions_large
    ,1,1,0,1,1,0)
colnames(snp_data_positions_large)[(dim(snp_data_positions_
    large)[2]-5):dim(snp_data_positions_large)[2]]=
c('p_H0_t_additivep_many','p_H0_t_additivep_few','p_H0_t_
    additivep_none',
  'p_H0_t_genoexactp_many','p_H0_t_genoexactp_few','p_H0_t_
      genoexactp_none')

snp_data_positions_large_temp_ct=array(as.integer(0),c(dim(
    merged_data)[1],6))
```

```
    snp_data_positions_large_temp_p=array(as.numeric(0),c(dim(
        merged_data)[1],6))

102 for(i_snp in 1:dim(merged_data)[1])
    {
      #t1 is case ct, t2 is ctrl ct
      t1=table(as.integer(merged_data[i_snp,train_set_cases]))[c(
          '0','1','2')]
      t1[is.na(t1)]=0
107   t2=table(as.integer(merged_data[i_snp,train_set_controls]))
          [c('0','1','2')]
      t2[is.na(t2)]=0

      #Now need to add new probabilities
      snp_data_positions_large_temp_ct[i_snp,1:6]=c(t1,t2)
112 t1=as.single(t1)
    t2=as.single(t2)
    if(sum(t1)+sum(t2)>0)
      {
        n_it=as.integer(1000)
117     p_H0_t_new=p_H0_t
        #includes new version
        snp_data_positions_large_temp_p[i_snp,1:2]=
          as.numeric(unlist(
            .Fortran("trend_prior_",t1,t2,K,p_H0,n_it,RR_high_max,
              p_H0_t,
122           as.integer(length(p_H0)),p_H0_t)[7]))
        n_it=as.integer(100)
        snp_data_positions_large_temp_p[i_snp,3:6]=
          as.numeric(unlist(
            .Fortran("genotype_prior_",t1,t2,K,p_H0,n_it,RR_high_
              max,p_H0_t,
127           as.integer(length(p_H0)),p_H0_t)[c(7,9)]))

      }

      if((i_snp%%10000)==0)
132     print(c(as.character(i_snp),as.character(dim(merged_data
          )[1]),date()))
    }

    snp_data_positions_large[,c('caseAA','caseAB','caseBB','
      ctrlAA','ctrlAB','ctrlBB')]=snp_data_positions_large_temp_
      ct
```

```
    snp_data_positions_large[,c('p_H0_t_additivep_many','p_H0_t_
        additivep_few',
137                                'p_H0_t_genoexactp_many_old','p_
                                       H0_t_genoexactp_few_old',
                                   'p_H0_t_genoexactp_many','p_H0_t_
                                       genoexactp_few')]=snp_data_
                                       positions_large_temp_p


    snp_data_positions_large[,'caseN']=
142   as.integer(snp_data_positions_large[,'caseAA'])+
      as.integer(snp_data_positions_large[,'caseAB'])+
      as.integer(snp_data_positions_large[,'caseBB'])
     snp_data_positions_large[,'ctrlN']=
      as.integer(snp_data_positions_large[,'ctrlAA'])+
147   as.integer(snp_data_positions_large[,'ctrlAB'])+
      as.integer(snp_data_positions_large[,'ctrlBB'])

    ### Need to add various multiple correction testing
        algorithms

152 remove(merged_data)
    gc(reset=T)

    training_set_available=test_set_concordant
    #NEW - keep only one 9p21 SNP
157 training_set_available[c(130233,130235,130236)]=F
    for(i_test in 1:2)
    {
      for(i_correct in 1:3)
      {
162     group_risk=naive_bayes_main_function(test_set,i_test,i_
            correct,group_risk)
      } #end i_correct
    } #end i_test

    remove(merged_data)
167 gc(reset=T)

    save(subject_CV,group_length,group_risk,file=paste(path,'
        thesis_work/6_A_Naive_Bayes/naive_bayes_r2_',r_squared_
        threshold,'_Cross_validation_',version_master[i_version],'
        _',CV_folds,'_folds_by_plate.Rdata',sep=""))
```

```
  } #end i_version
172
} #end fold/block
```

The code for 6_C_Naive_Bayes_Main_Function.r

```
path='c:/Documents and Settings/OHIUser/Desktop/Robbie/'
#path='C:/R/'
load(file=paste(path,'thesis_work/0_common_variables/common_
    variables.Rdata',sep=''))


#The central component of the Naive Bayes function
#Given the summary file, for a given test set, calculates Naive
    Bayes Scores
naive_bayes_main_function=function(test_set,i_test,i_correct,
    group_risk)
{
  #determine probability of being a case
  num_of_snps=dim(snp_data_positions_large)[1]
  #snp specific prob_case p(y=1|genotyped at SNP i)
  snp_data_temp_case=cbind(as.integer(snp_data_positions_large[,'
      caseAA'])
      ,as.integer(snp_data_positions_large[,'caseAB'])
      ,as.integer(snp_data_positions_large[,'caseBB']))
    snp_data_temp_ctrl=cbind(as.integer(snp_data_positions_large
      [,'ctrlAA'])
      ,as.integer(snp_data_positions_large[,'ctrlAB'])
      ,as.integer(snp_data_positions_large[,'ctrlBB']))
  prob_case_vector=(as.integer(snp_data_positions_large[,'caseAA'
      ])+
                    as.integer(snp_data_positions_large[,'caseAB'
                      ])+
                    as.integer(snp_data_positions_large[,'caseBB'
                      ]) )/
                   (as.integer(snp_data_positions_large[,'caseN'
                     ])+
                    as.integer(snp_data_positions_large[,'ctrlN'
                      ]))
  prob_ctrl_vector=1-prob_case_vector
  prob_case=(exp(mean(log(prob_case_vector[training_set_available
      ])))) #geometric mean
  prob_ctrl=1-prob_case
```

```
#moving on to defining risk based on genotype
#note that i_test is already available, giving which test is
    being performed
test_type=test_master[i_test]  # as in genoexactp vs additivep

#build genotype risk matrix depending on correction type
#an nSNP by 4 matrix where genotype_risk[i,j]=P(Y=1|X_i=j),
    where j=4 reprensts missing
if(test_type=='genoexactp')
{
  genotype_risk=array(as.numeric(0),c(num_of_snps,4))
  genotype_risk[,1]=as.integer(snp_data_positions_large[,'
      caseAA'])/(as.integer(snp_data_positions_large[,'ctrlAA'])
      +as.integer(snp_data_positions_large[,'caseAA']))
  genotype_risk[,2]=as.integer(snp_data_positions_large[,'
      caseAB'])/(as.integer(snp_data_positions_large[,'ctrlAB'])
      +as.integer(snp_data_positions_large[,'caseAB']))
  genotype_risk[,3]=as.integer(snp_data_positions_large[,'
      caseBB'])/(as.integer(snp_data_positions_large[,'ctrlBB'])
      +as.integer(snp_data_positions_large[,'caseBB']))
  genotype_risk[,4]=prob_case_vector
  genotype_risk[is.na(genotype_risk)]=max_risk
  genotype_risk[genotype_risk==0]=1-max_risk
  genotype_risk[genotype_risk>max_risk]=max_risk
  genotype_risk[genotype_risk<(1-max_risk)]=1-max_risk
  genotype_risk_ori=t(genotype_risk)
}

if(test_type=='additivep')
{
  ct_case=cbind(as.integer(snp_data_positions_large[,'caseAA'])
      ,
                as.integer(snp_data_positions_large[,'caseAB'])
                  ,
                as.integer(snp_data_positions_large[,'caseBB'])
                  )
  ct_ctrl=cbind(as.integer(snp_data_positions_large[,'ctrlAA'])
      ,
                as.integer(snp_data_positions_large[,'ctrlAB'])
                  ,
                as.integer(snp_data_positions_large[,'ctrlBB'])
                  )
  #note the following procedure uses matrix methods to
      calculate additive trend line slope and intercept for all
```

```
      SNPs at once! takes only a few seconds to calculate a
      quarter million slopes and intercepts!
   x=cbind(array(0,num_of_snps),array(1,num_of_snps),array(2,num
      _of_snps))
   sum_ct=(ct_case[,1]+ct_case[,2]+ct_case[,3]+ct_ctrl[,1]+ct_
      ctrl[,2]+ct_ctrl[,3])
   x_bar=(1/sum_ct) * rowSums( x * (ct_case+ct_ctrl))
   p_1_i=ct_case/(ct_case+ct_ctrl)
   p_1_i[is.na(p_1_i)]=1 #it doesn't matter anyway - NA -> their
       row sum is zero
   b= rowSums(
     ( (ct_case+ct_ctrl) * ( p_1_i - prob_case_vector) * (x-x_
        bar) ) ) /
   rowSums( (ct_case+ct_ctrl) * (x-x_bar) * (x-x_bar) )
   genotype_risk=prob_case_vector+b*(x-x_bar)
   genotype_risk=cbind(genotype_risk,prob_case_vector)
   genotype_risk[genotype_risk>max_risk]=max_risk
   genotype_risk[genotype_risk<(1-max_risk)]=1-max_risk
   genotype_risk_ori=t(genotype_risk)
}

#i_correct section
#this section requires resetting depending on the SNPs used, as
     during some corrections (ie Hochberg), most SNPs are
    removed if they are uninformative (p corrected = 1)

correct_type=correct_master[i_correct]
#generate false_positive values
false_positive_vector=snp_data_positions_large[,paste('p_H0_t_'
   ,test_type,'_',correct_type,sep='')]
true_positive_vector=1-false_positive_vector
#apply filter to make things run faster, depending on non-
   informative SNPs
#only need to consider true_positive_vector>0
relevant_snps=true_positive_vector>0 & training_set_available
relevant_snps_position=(1:length(relevant_snps))[relevant_snps]
true_positive_vector=true_positive_vector[relevant_snps]
false_positive_vector=false_positive_vector[relevant_snps]
genotype_risk=genotype_risk_ori[,relevant_snps]
prob_case_vector=prob_case_vector[relevant_snps]
prob_ctrl_vector=prob_ctrl_vector[relevant_snps]

#Add check for whether or not using OHGS Cross-Validation or
   test-sets
```

```
#If CV, then update only OHGS
#If test sets, update all (including OHGS for check only -
    without CV results should be comically good if everything
    runs properly)

if(length(test_set)==sum(test_set)) #ie test set is everything
    so run all cohorts
{
  i_subtrial_start=1
  i_subtrial_end=4
}
if(length(test_set)!=sum(test_set)) #test set is a subset so
    run only OHGS
{
  i_subtrial_start=3
  i_subtrial_end=3
}

#time to generate the Naive Bayes scores
for(i_subtrial in i_subtrial_start:i_subtrial_end)
{
  trial_subname=trial_submaster[i_subtrial]
  trial_name=trial_master[i_subtrial]

  #load SNP data
  remove(merged_data) #free up RAM for the next go around
  gc(reset=T) #shows current memory use
  load(file=paste(path,'thesis_work/5_merged_data/',trial_
      subname,'_merged_QC_r2_',r_squared_threshold,'_data.Rdata'
      ,sep=""))
  subject_class=array(0,dim(merged_data)[2]) #temporary
      storages for Naive Bayes Scores

  for(i_subject in 1:dim(merged_data)[2]) #for each person
  {
    if( F==(i_subtrial==3 & test_set[i_subject]==F)) #if OHGS
        CV, only run on CV test set
    {
      cur_snp_data=as.integer(merged_data[relevant_snps,i_
          subject]) # current SNP genotype information
      cur_genotype_risk=genotype_risk[4*(0:(-1+length(cur_snp_
          data)))+(cur_snp_data+1)] #risk across all SNPs based
          on genotype
      prob_case_given_genotype=prob_case_vector*false_positive_
```

```
            vector+cur_genotype_risk*true_positive_vector
          #added missing data filter #and add concordant filter
          missing_data=cur_snp_data==3 | test_set_concordant[
              relevant_snps_position]==F #remove missing values and
              SNPs which don't pass filter
          ### Calculation of the Naive Bayes Score!
126       subject_class[i_subject]=
            sum(log(prob_ctrl_vector[missing_data==F]/prob_case_
                vector[missing_data==F])) +
            sum( log ( 1/(1-prob_case_given_genotype[missing_data==
                F])-1  ) )
          ### End of calculation of Naive Bayes Score
        } #end check for test_set
131
        if((i_subject%%200)==0)
            print(paste('i_correct=',i_correct,', i_test=',i_test,'
                , i_subtrial=',i_subtrial,', i_subject_',i_subject,
              ', num_subjects=',dim(merged_data)[2],', ',date(),sep
                =''))
        #print update to screen about how many scores calculated,
            what time the calculation was completed, etc.
136   } #end current subject

      subject_class=subject_class+log(prob_case/prob_ctrl) #add p(y
          =1) - only useful for absolute comparisons

      #add scores to group_risk repository of Naive Bayes Scores
141   if(length(test_set)==sum(test_set))
        group_risk[i_correct,i_test,i_subtrial,1:length(subject_
            class)]=subject_class
      if(length(test_set)!=sum(test_set))
        group_risk[i_correct,i_test,i_subtrial,test_set]=subject_
            class[test_set]

146   #free up some space

    } #end i_subtrial - ie the cohort


151 return(group_risk)

} #end function
```

B.4 Posterior Probabilities

The following lists the code of `prior_scripts.f95`, which contains the Fortran code used to calculate the posterior probabilities used by Naive Bayes classifier.

The code for `prior_scripts.f95`.

```
   SUBROUTINE trend_prior(ct_case,ct_ctrl,K,p_H0,n_it,RR_high_max,
      p_H0_t,n_p_H0_length)
2  integer n_it, n_p_H0_length
   real ct_case(3), ct_ctrl(3), RR_high_max, K, p_H0(n_p_H0_length
      ), p_H0_t(n_p_H0_length)


   !---declare new variables
7  real x(3), a, R, S, N, maf
   real g(3), p(3), RR(3), f(3), q(3)
   real mu_1, sigma_1, mu_null, sigma_null, mu_0, sigma_0
   real p_t_H0, p_t_H1
   real p_RR_H1_temp, p_RR_H1_sum, num_U
12 !---counters
   integer i
   real i_real, n_it_real

   x(1)=0
17 x(2)=1
   x(3)=2

   !---flip if necessary
   if (ct_ctrl(1)<ct_ctrl(3)) then
22       a=ct_ctrl(1)
     ct_ctrl(1)=ct_ctrl(3)
     ct_ctrl(3)=a
     a=ct_case(1)
     ct_case(1)=ct_case(3)
27   ct_case(3)=a
   endif

   !---initialize some variables
   R=0
32 S=0
   N=0
   do 9 i=1,3
```

```
      R=R+ct_case(i)
     S=S+ct_ctrl(i)
37   N=N+ct_case(i)+ct_ctrl(i)
 9   continue



42  maf=0.5*(ct_ctrl(3)*2.+ct_ctrl(2))/(S)
    g(1)=(1-maf)**(2)
    g(2)=2*maf*(1-maf)
    g(3)=maf*maf
    f(1)=0
47  f(2)=0
    f(3)=0
    p_RR_H1_sum=0
    p_t_H1=0
    RR(1)=1
52
    !--- calculate null hypothesis
    RR(2)=1
    RR(3)=1
    f(1)=K/(g(1)*RR(1)+g(2)*RR(2)+g(3)*RR(3))
57  f(2)=f(1)*RR(2)
    f(3)=f(1)*RR(3)
    p(1)=f(1)*g(1)/K
    p(2)=f(2)*g(2)/K
    p(3)=f(3)*g(3)/K
62  q(1)=(1-f(1))*g(1)/(1.-K)
    q(2)=(1-f(2))*g(2)/(1.-K)
    q(3)=(1-f(3))*g(3)/(1.-K)
    mu_null=x(1)*( (S/N)*ct_case(1) - (R/N)*ct_ctrl(1) )+ &
        x(2)*( (S/N)*ct_case(2) - (R/N)*ct_ctrl(2) )+ &
67      x(3)*( (S/N)*ct_case(3) - (R/N)*ct_ctrl(3) )
    mu_0=mu_null - &
        R*S/N*( x(1)*(p(1)-q(1))+x(2)*(p(2)-q(2))+x(3)*(p(3)-q(3))
            )
    sigma_0=sqrt( &
      R*(S/N)**2* ( (x(1)*x(1)*p(1)+x(2)*x(2)*p(2)+x(3)*x(3)*p(3))
          - (x(1)*p(1)+x(2)*p(2)+x(3)*p(3) )**2) + &
72    S*(R/N)**2* ( (x(1)*x(1)*q(1)+x(2)*x(2)*q(2)+x(3)*x(3)*q(3))
          - (x(1)*q(1)+x(2)*q(2)+x(3)*q(3) )**2) )

    p_t_H0=1/(sigma_0*sqrt(2.*3.14159265358979323846264338327 95))*
       exp(-( mu_0 )**2/(2.*sigma_0**2))
```

```
      p_t_H1=0
77    p_RR_H1_sum=0

      !--- start loop
      do 13 i=1,n_it
        i_real=i
82      n_it_real=n_it
        if ( i_real/n_it_real<=0.5) then
             RR(3)=1./(RR_high_max-2.*(RR_high_max-1.)*(i_real/n_it_
                 real))
          RR(2)=1./(1.+(1./RR(3)-1.)*0.5)
            else
87            RR(3)=RR_high_max-2.*(RR_high_max-1.)*((n_it_real-i_
                 real)/n_it_real)
          RR(2)=0.5*(RR(3)-1.)+1.
        endif


92      f(1)=K/(g(1)*RR(1)+g(2)*RR(2)+g(3)*RR(3))
        f(2)=f(1)*RR(2)
        f(3)=f(1)*RR(3)
        p(1)=f(1)*g(1)/K
        p(2)=f(2)*g(2)/K
97      p(3)=f(3)*g(3)/K
        q(1)=(1-f(1))*g(1)/(1.-K)
        q(2)=(1-f(2))*g(2)/(1.-K)
        q(3)=(1-f(3))*g(3)/(1.-K)

102     mu_1= mu_null - &
          R*S/N*( x(1)*(p(1)-q(1))+x(2)*(p(2)-q(2))+x(3)*(p(3)-q(3))
              )
        sigma_1=sqrt( &
          R*(S/N)**2* (  (x(1)*x(1)*p(1)+x(2)*x(2)*p(2)+x(3)*x(3)*p
              (3)) - (x(1)*p(1)+x(2)*p(2)+x(3)*p(3) )**2) + &
          S*(R/N)**2* (  (x(1)*x(1)*q(1)+x(2)*x(2)*q(2)+x(3)*x(3)*q
              (3)) - (x(1)*q(1)+q(2)*p(2)+x(3)*q(3) )**2) )
107
        p_RR_H1_temp=(2.- 4.*abs( (i- ( (n_it+1.)/2. ) )/n_it))
           p_RR_H1_sum=p_RR_H1_sum+p_RR_H1_temp
        p_t_H1=p_t_H1 + &
          !---first is old p_t_RR
112       1/(sigma_1*sqrt(2.*3.14159265358979323846264338327 95))*exp
              (-( mu_1)**2/(2.*sigma_1**2)) * &
```

```
      p_RR_H1_temp
          !----second is p_RR_H1_temp unnormalized

   13   continue
117
   p_t_H1=p_t_H1/p_RR_H1_sum

   do 4 i=1,n_p_H0_length
     p_H0_t(i)=p_H0(i)*p_t_H0/(p_H0(i)*p_t_H0 + (1-p_H0(i))*p_t_H1
         )
122   4   continue



   return
127   end




132

   SUBROUTINE genotype_prior(ct_case,ct_ctrl,K,p_H0,n_it,RR_high_
      max,p_H0_t,n_p_H0_length,p_H0_t_new)
   integer n_it, n_p_H0_length
   real ct_case(3), ct_ctrl(3), RR_high_max, K, p_H0(n_p_H0_length
      ), p_H0_t(n_p_H0_length), p_H0_t_new(n_p_H0_length)

137   !---- general purpose variables
   real a, R, S, N
   integer n_it_1, n_it_2
   real n_it_1_real, n_it_2_real
   real RR_high_max_1, RR_high_max_2, g(3), f(3), RR(3), p(3), q
      (3)
142   real U(3)
   real RR_high_1, RR_high_2
   !---- new arrays
   real p_t_RR(n_it,n_it), p_RR_H1(n_it,n_it)
   real p_t_H1, maf, p_t_H0, p_RR_H1_sum, p_RR_H1_temp
147   !---- counters
   integer i_1, i_2, i
   real i_1_real, i_2_real
   real p_t_H0_new, p_t_H1_new
   real x, xp1, xp2, xp3, y
152
```

```
      n_it_1=n_it
      n_it_2=n_it

      p_t_H1=0
157         p_t_H1_new=0

      !----flip if necessary
      if (ct_ctrl(1)<ct_ctrl(3)) then
            a=ct_ctrl(1)
162     ct_ctrl(1)=ct_ctrl(3)
        ct_ctrl(3)=a
        a=ct_case(1)
        ct_case(1)=ct_case(3)
        ct_case(3)=a
167   endif


      RR_high_max_1=RR_high_max
      RR_high_max_2=RR_high_max
172


      n_it_1=n_it
      n_it_2=n_it
      n_it_1_real=n_it_1
177   n_it_2_real=n_it_2

      R=0
      S=0
      N=0
182   do 9 i=1,3
          R=R+ct_case(i)
        S=S+ct_ctrl(i)
        N=N+ct_case(i)+ct_ctrl(i)
       9  continue
187
      !---- get base variables

      maf=0.5*(ct_ctrl(3)*2.+ct_ctrl(2))/(S)
      g(1)=(1-maf)**(2)
192   g(2)=2*maf*(1-maf)
      g(3)=maf*maf
      f(1)=0
      f(2)=0
      f(3)=0
```

```
197   p_RR_H1_sum=0
      p_t_H1=0

      !--- go into two loops depending on ct_case(3)+ct_ctrl(3)
      if((ct_case(3)+ct_ctrl(3)).EQ.0) then
202     goto 201
      endif

      !--- start loop
      do 12 i_1=1,n_it_1
207     do 13 i_2=1,n_it_2
            i_1_real=i_1
          i_2_real=i_2

          if ( i_1/n_it_1_real<=0.5) then
212           RR_high_1=1./(RR_high_max_1-2.*(RR_high_max_1-1.)*(i_
                  1_real/n_it_1_real))
              else
              RR_high_1=RR_high_max_1-2.*(RR_high_max_1-1.)*((n_it_
                  1_real-i_1_real)/n_it_1_real)
          endif

217           if ( i_2/n_it_2_real<=0.5) then
              RR_high_2=1./(RR_high_max_2-2.*(RR_high_max_2-1)*(i_2
                  _real/n_it_2_real))
            else
              RR_high_2=RR_high_max_2-2.*(RR_high_max_2-1.)*((n_it_
                  2_real-i_2_real)/n_it_2_real)
          endif
222
          RR(1)=1
          RR(2)=RR_high_1
          RR(3)=RR_high_2


227
          f(1)=K/(g(1)*RR(1)+g(2)*RR(2)+g(3)*RR(3))
          f(2)=f(1)*RR(2)
          f(3)=f(1)*RR(3)
          p(1)=f(1)*g(1)/K
232       p(2)=f(2)*g(2)/K
          p(3)=f(3)*g(3)/K
          q(1)=(1-f(1))*g(1)/(1.-K)
          q(2)=(1-f(2))*g(2)/(1.-K)
          q(3)=(1-f(3))*g(3)/(1.-K)
```

```
237
          U(1)=(p(1)*R+q(1)*S)*( ct_case(1)/(ct_case(1)+ct_ctrl
             (1)) - p(1)*R/(p(1)*R+q(1)*S))**2/ ((R/N)*(S/N))
          U(2)=(p(2)*R+q(2)*S)*( ct_case(2)/(ct_case(2)+ct_ctrl
             (2)) - p(2)*R/(p(2)*R+q(2)*S))**2/ ((R/N)*(S/N))
          U(3)=(p(3)*R+q(3)*S)*( ct_case(3)/(ct_case(3)+ct_ctrl
             (3)) - p(3)*R/(p(3)*R+q(3)*S))**2/ ((R/N)*(S/N))
242
          p_RR_H1_temp=(2.- 4.*abs( (i_1_real- ( (n_it_1_real+1.)
             /2. ) )/n_it_1_real)) * &
                   (2.- 4.*abs( (i_2_real- ( (n_it_2_real+1.)/2.
                      ) )/n_it_2_real))
       p_RR_H1_sum=p_RR_H1_sum+p_RR_H1_temp
       p_t_H1=p_t_H1 + &
247      !---first is old p_t_RR
            (1./sqrt(2.*3.14159265358979323846264338327 95)* &
         exp(-(   (sqrt(U(1))+sqrt(U(2))+sqrt(U(3)))/3.*sqrt(3.)
            )**2/2.)) * p_RR_H1_temp
            !---second is p_RR_H1_temp unnormalized
         xp1=( R*p(1) )  /  ( R*p(1)+S*q(1) )
252      xp2=( R*p(2) )  /  ( R*p(2)+S*q(2) )
         xp3=( R*p(3) )  /  ( R*p(3)+S*q(3) )

         x=(  ct_case(1) / (ct_case(1)+ct_ctrl(1)) -xp1 )**2 /
              (1/(ct_case(1)+ct_ctrl(1)) * (xp1) * (1-xp1)   ) +
            &
           (  ct_case(2) / (ct_case(2)+ct_ctrl(2)) -xp2 )**2 /
              (1/(ct_case(2)+ct_ctrl(2)) * (xp2) * (1-xp2)   )
             + &
257        (  ct_case(3) / (ct_case(3)+ct_ctrl(3)) -xp3 )**2 /
              (1/(ct_case(3)+ct_ctrl(3)) * (xp3) * (1-xp3)   )
!      p_t_H1_new=p_t_H1_new + 0.5*x*exp(-x/2.0) * p_RR_H1_temp
       p_t_H1_new=p_t_H1_new + 0.5*exp(-x/2.0) * p_RR_H1_temp


262 !        if ( 0.5*x*exp(-x/2.0) <= y ) then
    !        p_H0_t(1)=0.5*x*exp(-x/2.0)
    !        p_H0_t(2)=y
    !        p_H0_t_new(1)=RR_high_max_2-2.*(RR_high_max_2-1.)*((n_
       it_2_real-i_2_real)/n_it_2_real)
    !        p_H0_t_new(2)=RR(3)
267 !        return
    !      endif
```

```
     13    continue
     12 continue
272
    p_t_H1=p_t_H1/p_RR_H1_sum
    p_t_H1_new=p_t_H1_new/p_RR_H1_sum


277 !--- do normal
    RR(1)=1
    RR(2)=1
    RR(3)=1
    f(1)=K/(g(1)*RR(1)+g(2)*RR(2)+g(3)*RR(3))
282 f(2)=f(1)*RR(2)
    f(3)=f(1)*RR(3)
    p(1)=f(1)*g(1)/K
    p(2)=f(2)*g(2)/K
    p(3)=f(3)*g(3)/K
287 q(1)=(1-f(1))*g(1)/(1-K)
    q(2)=(1-f(2))*g(2)/(1-K)
    q(3)=(1-f(3))*g(3)/(1-K)
        U(1)=(p(1)*R+q(1)*S)*( ct_case(1)/(ct_case(1)+ct_ctrl(1)) -
             p(1)*R/(p(1)*R+q(1)*S))**2/ ((R/N)*(S/N))
        U(2)=(p(2)*R+q(2)*S)*( ct_case(2)/(ct_case(2)+ct_ctrl(2)) -
             p(2)*R/(p(2)*R+q(2)*S))**2/ ((R/N)*(S/N))
292     U(3)=(p(3)*R+q(3)*S)*( ct_case(3)/(ct_case(3)+ct_ctrl(3)) -
             p(3)*R/(p(3)*R+q(3)*S))**2/ ((R/N)*(S/N))
        p_t_H0=1/sqrt(2.*3.14159265358979323846264338327 95)*exp(-(
              (sqrt(U(1))+sqrt(U(2))+sqrt(U(3)))/3.*sqrt(3.)  )**2/
            2.)

        xp1=( R*p(1) )  /  ( R*p(1)+S*q(1) )
        xp2=( R*p(2) )  /  ( R*p(2)+S*q(2) )
297     xp3=( R*p(3) )  /  ( R*p(3)+S*q(3) )

        x=(  ct_case(1) / (ct_case(1)+ct_ctrl(1)) -xp1 )**2 /   (1/
           (ct_case(1)+ct_ctrl(1)) * (xp1) * (1-xp1)   ) + &
          (  ct_case(2) / (ct_case(2)+ct_ctrl(2)) -xp2 )**2 /   (1/
             (ct_case(2)+ct_ctrl(2)) * (xp2) * (1-xp2)   ) + &
          (  ct_case(3) / (ct_case(3)+ct_ctrl(3)) -xp3 )**2 /   (1/
             (ct_case(3)+ct_ctrl(3)) * (xp3) * (1-xp3)   )
302
!   p_t_H0_new= 0.5*x*exp(-x/2.0)
    p_t_H0_new= 0.5*exp(-x/2.0)
```

```
!---- end normal
307
      do 2 i=1,n_p_H0_length
        p_H0_t(i)=p_H0(i)*p_t_H0/(p_H0(i)*p_t_H0 + (1-p_H0(i))*p_t_H1
            )
        p_H0_t_new(i)=p_H0(i)*p_t_H0_new/(p_H0(i)*p_t_H0_new + (1-p_
            H0(i))*p_t_H1_new)
312   2    continue

      return

      201 do 14 i_1=1,n_it_1
317     do 15 i_2=1,n_it_2
            i_1_real=i_1
          i_2_real=i_2

          if ( i_1/n_it_1_real<=0.5) then
322           RR_high_1=1./(RR_high_max_1-2.*(RR_high_max_1-1.)*(i_
                  1_real/n_it_1_real))
              else
              RR_high_1=RR_high_max_1-2.*(RR_high_max_1-1.)*((n_it_
                  1_real-i_1_real)/n_it_1_real)
          endif

327           if ( i_2/n_it_2_real<=0.5) then
                RR_high_2=1./(RR_high_max_2-2.*(RR_high_max_2-1)*(i_2
                    _real/n_it_2_real))
              else
                RR_high_2=RR_high_max_2-2.*(RR_high_max_2-1.)*((n_it_
                    2_real-i_2_real)/n_it_2_real)
          endif
332
          RR(1)=1
          RR(2)=RR_high_1
          RR(3)=RR_high_2


337
          f(1)=K/(g(1)*RR(1)+g(2)*RR(2)+g(3)*RR(3))
          f(2)=f(1)*RR(2)
          f(3)=f(1)*RR(3)
          p(1)=f(1)*g(1)/K
342       p(2)=f(2)*g(2)/K
```

```
      p(3)=f(3)*g(3)/K
      q(1)=(1-f(1))*g(1)/(1.-K)
      q(2)=(1-f(2))*g(2)/(1.-K)
      q(3)=(1-f(3))*g(3)/(1.-K)
347
          U(1)=(p(1)*R+q(1)*S)*( ct_case(1)/(ct_case(1)+ct_ctrl
             (1)) - p(1)*R/(p(1)*R+q(1)*S))**2/ ((R/N)*(S/N))
          U(2)=(p(2)*R+q(2)*S)*( ct_case(2)/(ct_case(2)+ct_ctrl
             (2)) - p(2)*R/(p(2)*R+q(2)*S))**2/ ((R/N)*(S/N))
          U(3)=0
352
      p_RR_H1_temp=(2.- 4.*abs( (i_1_real- ( (n_it_1_real+1.)/2.
         ) )/n_it_1_real)) * &
                   (2.- 4.*abs( (i_2_real- ( (n_it_2_real+1.)/2.
                      ) )/n_it_2_real))
      p_RR_H1_sum=p_RR_H1_sum+p_RR_H1_temp
      p_t_H1=p_t_H1 + &
357     !---first is old p_t_RR
            (1/sqrt(2.*3.14159265358979323846264338327 95)* &
        exp(-(    (sqrt(U(1))+sqrt(U(2))+sqrt(U(3)))/2.*sqrt(2.)
           )**2/2.)) * p_RR_H1_temp
            !---second is p_RR_H1_temp unnormalized

362       xp1=( R*p(1) )  /  ( R*p(1)+S*q(1) )
          xp2=( R*p(2) )  /  ( R*p(2)+S*q(2) )

          x=(  ct_case(1) / (ct_case(1)+ct_ctrl(1)) -xp1 )**2 /
               (1/(ct_case(1)+ct_ctrl(1)) * (xp1) * (1-xp1)    ) +
             &
            (  ct_case(2) / (ct_case(2)+ct_ctrl(2)) -xp2 )**2 /
                 (1/(ct_case(2)+ct_ctrl(2)) * (xp2) * (1-xp2)    )
367 !     p_t_H1_new=p_t_H1_new + 0.5*x*exp(-x/2.0) * p_RR_H1_temp
      p_t_H1_new=p_t_H1_new + 0.5*exp(-x/2.0) * p_RR_H1_temp


    15   continue
372   14 continue
    p_t_H1=p_t_H1/p_RR_H1_sum
    p_t_H1_new=p_t_H1_new/p_RR_H1_sum


377 !--- do normal
    RR(1)=1
```

```
      RR(2)=1
      RR(3)=1
      f(1)=K/(g(1)*RR(1)+g(2)*RR(2)+g(3)*RR(3))
382   f(2)=f(1)*RR(2)
      f(3)=f(1)*RR(3)
      p(1)=f(1)*g(1)/K
      p(2)=f(2)*g(2)/K
      p(3)=f(3)*g(3)/K
387   q(1)=(1-f(1))*g(1)/(1-K)
      q(2)=(1-f(2))*g(2)/(1-K)
      q(3)=(1-f(3))*g(3)/(1-K)
          U(1)=(p(1)*R+q(1)*S)*( ct_case(1)/(ct_case(1)+ct_ctrl(1)) -
               p(1)*R/(p(1)*R+q(1)*S))**2/ ((R/N)*(S/N))
          U(2)=(p(2)*R+q(2)*S)*( ct_case(2)/(ct_case(2)+ct_ctrl(2)) -
               p(2)*R/(p(2)*R+q(2)*S))**2/ ((R/N)*(S/N))
392       U(3)=0
          p_t_H0=1/sqrt(2.*3.14159265358979323846264338327 95)*exp(-(
               (sqrt(U(1))+sqrt(U(2))+sqrt(U(3)))/2.*sqrt(2.) )**2/
              2.)
          xp1=( R*p(1) ) / ( R*p(1)+S*q(1) )
          xp2=( R*p(2) ) / ( R*p(2)+S*q(2) )

397       x=( ct_case(1) / (ct_case(1)+ct_ctrl(1)) -xp1 )**2 / (1/
             (ct_case(1)+ct_ctrl(1)) * (xp1) * (1-xp1) ) + &
            ( ct_case(2) / (ct_case(2)+ct_ctrl(2)) -xp2 )**2 / (1/
               (ct_case(2)+ct_ctrl(2)) * (xp2) * (1-xp2) )

!     p_t_H0_new= 0.5*x*exp(-x/2.0)
      p_t_H0_new= 0.5*exp(-x/2.0)
402
      !--- end normal


      !--- end normal
407


      do 3 i=1,n_p_H0_length
        p_H0_t(i)=p_H0(i)*p_t_H0/(p_H0(i)*p_t_H0 + (1-p_H0(i))*p_t_H1
            )
        p_H0_t_new(i)=p_H0(i)*p_t_H0_new/(p_H0(i)*p_t_H0_new + (1-p_
            H0(i))*p_t_H1_new)
412   3   continue
```

```
417    return
       end
```