Title
Finding Genes Related to Disease Using Statistical Learning

Author
Goldstein, Benjamin Alan

Publication Date
2011

Peer reviewed|Thesis/dissertation

Finding Genes Related to Disease Using Statistical Learning

by

Benjamin Alan Goldstein

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Biostatistics

and the Designated Emphasis

in

Computational and Genomic Biology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Alan Hubbard, Chair
Professor Lisa Barcellos
Professor Mark van der Laan

Spring 2011

Finding Genes Related to Disease Using Statistical Learning

Abstract

Finding Genes Related to Disease Using Statistical Learning

by

Benjamin Alan Goldstein
Doctor of Philosophy in Biostatistics
and the Designated Emphasis in Computational and Genomic Biology

University of California, Berkeley

Professor Alan Hubbard, Chair

This dissertation consists of the analyses of three separate genetic association datasets. Each represents a unique data structure with a different question of interest that therefore require distinct approaches and methodologies. As such, the three substantive chapters (2-4) can each stand on their own. However, the over-arching question in each of these studies is the same: which genes (or genetic material) are related to the disease or outcome being studied. Moreover, while the methodologies are each distinct, they all incorporate statistical learning methodologies to obtain some modicum of inference.

Study 1 - As computational power has improved the application of statistical learning algorithms to finding SNPs related to disease has become more ubiquitous. The hope is that these algorithms will be more capable than typical marginal testing in detecting SNPs with higher order effects. The Random Forests (RF) algorithm is one such algorithm that has seen increased use with genetic data. As part of its output, RF ranks the predictor variables (SNPs) on their relative importance. The present study represents the first application of the RF algorithm to Genome Wide Association (GWA) data and investigates how best to use the algorithm for this unique data structure. A multiple sclerosis (MS) GWA data set is used for the analysis. Results indicate the typical tuning parameter settings need to be adjusted for the high degree of sparsity in the data. Furthermore, most meaningful results were obtained when both unimportant and *overly* important SNPs were removed. RF was able to replicate some previous findings using the same data. Moreover, four genes not previously associated with MS were identified.

Study 2 - In many analyses, one has data on one level but desires to draw inference on another level. For example, in genetic association studies, one observes units of DNA referred to as SNPs, but wants to determine whether genes that are comprised of SNPs are associated with disease. While there are some available approaches for addressing this issue, they usually involve making parametric assumptions and are not easily generalizable. A statistical test is proposed for testing the association of

a set of variables with an outcome of interest. No assumptions are made about the functional form relating the variables to the outcome. A general function is fit using any statistical learning algorithm, with the SuperLearner algorithm suggested. The parameter of interest is the cross-validated risk and this is compared to an expected risk. A Wald test is proposed using the influence curve of the cross-validated risk to obtain the variance. It is shown both theoretically and via simulation that the test maintains appropriate type I error control and is more powerful than parametric tests under more general alternatives. The test is applied to an MS candidate gene study. Three separate analyses are performed highlighting the flexibility of the approach.

Study 3 - Secondary analyses, such as Gene Ontology and Motif analysis, have become central components of gene expression experiments, allowing researchers to derive biological understanding from the set of genes that are differentially expressed. An important statistical task is determining which genes should be passed on to such programs and how the genes should be grouped for analysis. The typical approach is to cluster the set of differentially expressed genes, and pass these clusters on to the secondary analyses. However, many expression experiments have specific hypotheses which allow one to analyze the genes and group them in a more targeted approach. To illustrate the utility of being more specific, a gene expression study of *C. elegans* is used where a particular outcome was observed and hoped to be explained. A general model is fit and analyzed to estimate the parameters corresponding to the specific hypothesis, leading to four natural groupings of the differentially expressed genes. These groupings lead to meaningful results in the secondary analyses that allow for the biologist to make robust hypotheses that are experimentally confirmed. It is shown that a traditional approach would not have yielded such robust findings.

Dedicated in memory to Grandpa Calvin who was great with numbers
and Grandpa Moe who always wanted to go back to school

Contents

List of Figures

List of Tables

Acknowledgments

I enjoy statistics because it cannot be done in vacuum. Instead, good statistics only occurs through close work and collaboration with others who have expertise in other domains. With that in mind there are many people I'd like to thank and acknowledge. First and foremost are my advisors Alan Hubbard and Lisa Barcellos. I have been fortunate to have been advised by two wonderful and caring professors. I appreciate not only their technical expertise but the personal relationships I have developed with them. They have both served as a sounding board for both life and academic decisions and I expect that they will continue to do so for years to come.

I have been fortunate to collaborate with some amazingly intelligent people. The members of the Genetic Epidemiology Research Group have been a source of both intellectual and social support. Farren Briggs has become both a wonderful collaborator and good friend. Gary Artim forced me to use UNIX, C & PERL and I have learned more from him than I have from many professors. I am indebted to Adele Cutler for all of her help with the Random Forests work. The gene expression work would not have been possible without the collaborators at the Buck Institute. Di Chen was the biologist in charge, and I am convinced the success of the project is much more a reflection on him than me.

Each professor I have been exposed to has a unique approach to statistics and I have done my best to absorb what I can from each of them. It is impossible to take a class from Mark van der Laan and not change the way you think about data. I thank him for chairing my qualifying exam and reading this dissertation. I wish I could have worked even more with him. I want to thank Bin Yu for introducing me to statistical learning, serving on my qualifying exam committee, and challenging my ideas. Like many students before me I find myself constantly reflecting on the lessons of David Freedman. I feel incredibly fortunate to have taken two classes with him before he passed away.

School is defined by students, and I have had some great peers. Particular thanks goes to Eric Polley, Ori Stitelman and Johann Gagnon-Bartsch, all great statisticians who have listened to my ideas and complaints and have helped push me along. They are more than colleagues, they are friends.

Finally I'd like to thank my family. They have always been supportive of me even if they didn't understand what I was doing or why I was doing it. Their love and support has been invaluable. Ben Allen has been a friend since college and has become as close as family as one can be. Finally, there is no way I can sufficiently thank my wife, Cheng. I would not be where I am today and would not know where I was going without you. You have served as my support and inspiration since I met you. And of course Puma - you kept me company while writing, and I solved many problems during our walks together.

This research was funded in part by the Russell M. Grossman Endowment and a National Institutes of Health NRSA Trainee appointment on grant T32 HG 00047.

Chapter 1

Introduction

The following pages represent much of the work I have performed over the past three and half years as a doctoral student. These chapters represent three distinct papers that can easily stand on their own, each motivated from a different dataset. I believe the methods one uses should be motivated by the data one has and the questions one wants to answer. For this reason, each chapter consists of very different primary methods. However, taken together they represent my interest in and approach to statistics. The three chapters can also be read as illustrating what I see as three of the primary roles of a biostatistician: developing new methods (Chapter 3), evaluating existing methods (Chapter 2) and analyzing data (Chapter 4).

The common question overriding most of this work is identifying which genes are related to an outcome (usually disease) in large genetic studies. To answer this question I have focused on the use of computational tools, particularly statistical learning algorithms[1]. These projects have allowed me to bridge two sides of statistics: inference & prediction. While these genetic studies are exploratory in nature, the questions asked in the following pages are primarily ones of inference: which genes are related to disease? However, due to the complexity (both biological and statistical) of genetic data, typical inferential tools are ill suited for answering this question. This is where statistical learning comes in. These methods allow the user to flexibly search through the data, finding complex relationships that provide insight into the data problem. They are often capable of handling the high dimensional and complex data encountered in genetics. While most of these algorithms were developed within the domain of computer science, statisticians have been able to provide great insight into their properties. I hope that the following pages continue that line of insight.

[1]I use the term "statistical learning" distinctly from the more common "machine learning." From my perspective the two refer to the same set methodologies: computational algorithms aimed at searching for the best relationship in the data. The distinction applies to the approach taken by the user. Using these algorithms as a "black box" is the domain of machine learning. Delving into and trying to understand the statistical properties of these algorithms and optimizing them in turn is statistical learning.

Chapter 2 (along with the appendix) illustrates my work with the Random Forests (RF) algorithm. This was my first project as a doctoral student and the task was simply to determine whether the algorithm could be used with genome wide association (GWA) studies. GWA studies are characterized by having 100's of thousands of units of genetic information for thousands of people. Adding to the difficulty of the problem, it is presumed that most of the genetic data is not related to the outcome of interest. At the time no one had attempted to use this (or any) algorithm with this large a dataset. However, with advanced computational ability it has become feasible. The general conclusion from the work with RF is that, yes, meaningful information can be extracted from GWA data. However, unsurprisingly, the typical approach to using the algorithm needed to be changed - suggestions for these changes are provided. Chapter 2 represents the published work on this topic. Since I found the theory behind RF particularly interesting, I have written about this in the appendix. This section provides insight into the value added when one understands the algorithms that they use. These two sections do not represent all of my work on RF. In fact, I still consider this an unfinished project. My hope was to derive statistical properties for the variable importance measures to aid in variable selection (i.e. inference). However, after much effort, the work proved fruitless and I had to conclude that either (a) none existed or (b) they did exist but I was incapable of discerning them. The gigabytes of simulation output serve as a reminder that not all projects end in pretty papers.

While the RF project was "given" to me, Chapter 3 on gene based tests arose more organically. Like all good projects, there are multiple roots to its formulation. The idea began with my work on RF, where I wanted to ask the simple question of whether the prediction I was getting was "good" i.e. better than what I'd expect by chance. My thought was that if it wasn't, then one should not really look at the variable importance measures. Later, I was at the American Society for Human Genetics conference where the topic of the day was rare variant analysis. Since analyzing individual rare variants lacks statistical power, the advocated approach was to sum up all the rare variants in a gene. While this made sense for rare variants it forced me to consider, why not do this for all SNPs in a gene. Furthermore, instead of just summing up the SNPs, why not combine them in a flexible manner as one would do with a statistical learning algorithm. These thoughts grew into the work of Chapter 3.

While the initial method was motivated by genetic data, it also became quite clear to me that it was generally applicable outside of genetics. For this reason I have taken a very general approach with the development of the test statistic. However, it is within genetics that the applications are particularly diverse. The flexibility of the method allows one to ask (and hopefully answer) many subtle and interesting questions. I have tried to illustrate this with three different and interesting questions, all derived from the same dataset.

The final chapter is a departure from the work of the previous two, primarily

because I was working with very different data. To expand my exposure, I began a collaboration with biologists studying gene expression in model organisms (i.e. *C. Elegans*, mice, drosophila). The aim of the lab was to find genes involved in extended lifespan. Chapter 4, represents the results of one of these analyses that was particularly successful. Due to the nature of the experiment and the particular question asked, I was able to analyze the data in a targeted way as opposed to the more general approaches typical of these studies. A particularly unique aspect of working with model organisms, is that one is able to follow-up on the derived results and confirm them experimentally. For a statistician this is both exciting and intimidating. It is rare that our analyses can be confirmed to be correct, but also rare that they can be definitively invalidated. Fortunately many of the proposed genes were validated. While methodologically most distinct, at its core, this work illustrates the power of asking targeted and specific questions about ones data, present in the earlier chapters.

In total, the following is about what questions do we want to ask of the data and how can we best answer them. For a science, statistics has remarkable space for creativity. I find that the best work occurs when one thinks about the data creatively. I hope the follow pages illustrate this concept.

Chapter 2

Approaches and Considerations for Applying Random Forests to Genome-Wide Association Studies

2.1 Background

Genome-wide association (GWA) studies are a well-established approach for identifying genetic regions of interest for many common complex diseases and traits [WTCCC, 2007]. These studies are characterized by examining genetic information from thousands of individuals, at hundreds of thousands of loci across the human genome known as single nucleotide polymorphisms (SNPs). The standard assumption is that either variation at particular loci leads to changes in biological function, which in turn leads to disease, or that associated loci are in linkage disequilibrium (LD) with other disease causing variants. By examining genotypes derived from individuals with and without the disease or trait of interest, one can discern such variation. This is typically done by performing a marginal chi-square test with some control for multiple testing. However, since each causal SNP will confer risk under an unknown and different genetic model (i.e. additive, dominant, recessive), and may also interact with other SNPs (epistasis), a marginal test will be a less successful approach for finding the association [Heidema et al., 2006]. Ideally, one would simply test all possible genetic models of association, including those for interaction. However, in the context of a GWA study, this is not computationally feasible.

Recent emphasis has been on the use of machine learning techniques to identify potential causal variants. Such techniques include logic regression [Kooperberg and Ruczinski, 2005], multi-dimensional reduction (MDR) [Motsinger and Ritchie, 2006], support vector machines (SVM) [Yoon et al., 2003], and Random Forests (RF) [Bureau et al., 2005]. While these techniques are each unique, they have a shared characteristic whereby each algorithm searches over a transformed version of the fea-

ture space attempting to find the optimal solution to the problem while minimizing some empirical risk. Importantly, the algorithms make minimal assumptions about the causal mechanism. This means these algorithms may be more suited for identifying variants where the causal mechanism is unknown and complex, as is the case with complex genetic diseases.

Each of these methods has utility for finding structure in genetic data, where the best algorithm will depend on the true nature of the underlying association. However, the focus of the current study is RF because of the ability of this method to identify variables of interest from very large datasets. Equally important, RF is a relatively straightforward algorithm, both to understand and interpret. Unsurprisingly, there has been a slow but steady use of RF in the genomic literature since its introduction in 2001 [Bureau et al., 2005, Díaz-Uriarte and Alvarez de Andrés, 2006, Glaser et al., 2007, Lunetta et al., 2004, Meng et al., 2009, Nonyane and Foulkes, 2008, Sun et al., 2007].

RF was first introduced by Leo Breiman [Breiman, 2001] and is a natural extension of his previous work on classification and regression trees (CART) [Breiman et al., 1984] and bootstrap aggregating (or bagging) [Breiman, 1996a]. CART is an effective tool for building a classifier, but tends to be data dependent, where even small data changes can result in different tree structures. Bagging is a process whereby data are sampled with replacement and the classifier is grown using this bootstrap sample. After many iterations, results are aggregated over all trees to create a less variable classifier with a lower prediction error when compared to the original classifier. In bagging, the variance reduction is limited by the correlation between trees; as correlation is decreased or minimized, the potential for reduction is increased. The RF algorithm (see Figure 2.1) begins by bagging CART trees. To reduce the correlation between trees, instead of searching over all p variables at each node for the optimal split, a search is performed over a random subset, $m \leq p$, at each node. The algorithm continues to split the data until no further splits are possible, either because the node is pure (all of one class), or there are no more variables upon which to split. While the CART algorithm calls for the tree to be pruned for increased stability, RF leaves the tree unpruned, as bagging is used to decrease the variance created by the lack of pruning.

An aspect of the bagging procedure is that a natural, internal error rate is created. Within each bootstrap sample, approximately 37% of the original data will be unselected, referred to as the out-of-bag (OOB) sample [Breiman, 1996b]. RF passes OOB samples down the tree to obtain a class prediction. After the full forest is grown, the class predictions are compared to the true classes generating the out-of-bag error rate (OOB-ER). This error-rate can be used to compare the prediction accuracy of one set of inputs to another, behaving similarly to cross-validation [Hastie et al., 2009].

An appeal of RF is that the forest of trees contains a large amount of information about the relationship between the variables and observations. This information can be used for prediction, clustering, imputing missing data, and detecting outliers. Of

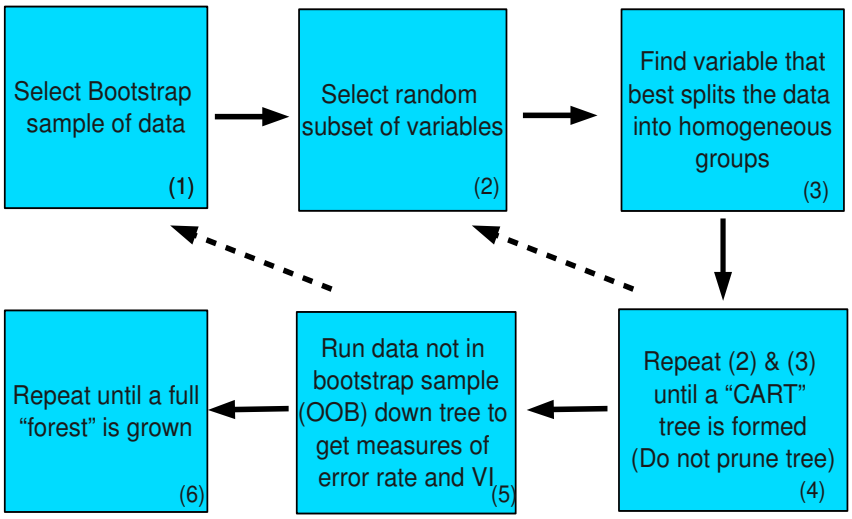


Figure 2.1: The RF algorithm begins by selecting a bootstrap sample of the data (1). A random subset of the variables is selected (2) and searched over to find the optimal split (3). This is repeated until an unpruned CART tree is formed (4). The data not part of the bootstrap sample is run down the tree to derive the error rate and measures of VI (5). This is repeated until a full forest is grown (6).

great interest to genetic epidemiologists, is the ability of RF to identify important variables. After each OOB sample is passed down the tree to produce a prediction error for the sample, one then permutes each variable in the tree across samples, and passes the same observation down the tree again. Any increase in misclassification helps determine the importance of that variable. This type of variable importance (VI) can be derived from disparate variable types (categorical, ordinal, continuous), and makes no assumptions about the data generating distribution for the outcome. However, unlike a formal hypothesis, it is best to consider the output of a RF analysis as a rank ordering of important variables worthy of further investigation, not as a list of variables with a known Type I error rate. Moreover, RF VI is best suited for identifying regions of interest as opposed to actual causal variants, where other more targeted methods are preferable [Bembom et al., 2007].

Utilization of RF requires choosing between three tuning parameters: (1) number

of trees to grow (*ntree*), (2) number of variables to select per-node (*mtry*), and in the case of classification, (3) class weights. While most applications in the literature have successfully implemented RF using default settings, applying RF to large GWA datasets is more complicated. Few studies have examined the various tuning parameters. The two most comprehensive reviews concluded that RF predictions were stable and robust to small fluctuations in tuning parameters settings, but often there were optimal settings [Díaz-Uriarte and Alvarez de Andrés, 2006, Genuer et al., 2008]. While both studies provide useful information, the largest dataset examined by each contained only 9,868 predictors and 78 observations. This is obviously much smaller than the data analyzed in a typical GWA study.

Further complicating RF analysis, beyond the large feature space, is that GWA data tend to be highly correlated, with potentially, many regions of LD among SNPs. Also, the data are assumed to be highly sparse, meaning there is an apriori assumption that the vast majority of SNPs will not be associated with the disease. While many of these issues have been discussed in the literature, none have been considered in the context of a large GWA dataset. Moreover, many of the strategies one would employ with smaller data sets (e.g. permutation, cross-validation etc.) are not feasible due to computational constraints. Instead of working with simulated data which can be less realistic, we investigated the application of RF using a large multiple sclerosis (MS) GWA study dataset comprised of cases and controls. The aim of the current study was two-fold: (1) to illustrate how one would go about tuning RF for a particular GWA analysis, and (2) to determine whether RF would duplicate results found in the original MS GWA study, as well as identify any new loci of interest.

2.2 Methods

2.2.1 Genotypes

Data were derived from a 2007 MS case-control study conducted by the International Multiple Sclerosis Genetics Consortium [Hafler et al., 2007] and were comprised of genotypes for a total of 325,807 SNPs (Affymetrix GeneChip Human Mapping 500K array) in 931 MS cases and 2,431 controls (n = 3,362). Stringent quality control (QC) analyses were applied to the dataset as previously described, including the removal of population outliers [Hafler et al., 2007]. SNPs with greater than 10% missing data were removed. The genetic inflation factor was 1.06, indicating negligible population stratification [Hafler et al., 2007].

Less than 1% of the genetic data contained missing values. There are a few different ways missing data can be handled within RF. However, since the data were derived from a dense SNP marker panel and had minimal missingness, any missing values were imputed with Beagle 2.13 [Browning and Browning, 2007]. Allelic data were then recoded into genotype format using PLINK 1.05 [Purcell et al., 2007],

producing three categories for each SNP (0, 1 and 2 copies of the minor allele). Since the optimal binary split is found at each node, this allows for the algorithm to be agnostic to recessive, dominant or additive effects. An allelic chi-square test (df = 1) was performed to calculate marginal associations for comparison.

2.2.2 RF Implementation

The RF code was originally written in Fortran by Breiman and Cutler. There is also an R package randomForest based on the same Fortran code [Liaw and Wiener, 2002]. Neither implementation could be used for the large GWA dataset in the current study. The original RF code has been licensed to Salford Systems[Sal], and they recently optimized the Fortran version, v.6.4.0.179, for application to large datasets. In preliminary testing of small datasets, similar results were found between the three implementations of RF (data not shown). RF was implemented in a server environment with 8 2/GHz cpus and 32GB of memory. Run time was dependent on data size and *mtry*, ranging from a few seconds per tree to over 10 minutes per tree ($\sim$ 1 week for a full forest).

2.2.3 Tuning Parameters Considered

Number of variables to choose per node (mtry)

The primary tuning parameter in RF is the number of variables to search at each node (*mtry*). This parameter controls the bias-variance trade-off. Searching over fewer variables per node will produce less correlated trees, reducing the overall variance of the prediction. However, this will also decrease the accuracy of each individual tree, increasing the bias. The *mtry* can also be viewed as controlling the complexity of the model, with a smaller value leading to a more complex, less sparse solution (see below). Breiman originally suggested choosing the $\text{int}(\log_2 p + 1)$ of the number of predictors per node. In the R implementation, the default value is the square root of the number of predictors.

For a GWA dataset, this would entail examining approximately 550 SNPs per node. As noted by Breiman, when there are many weak predictors, this number may need to be increased. It has also been noted that *mtry* is more important for VI calculation than for prediction, and that with sparse data, *mtry* = p leads to greatest stability [Genuer et al., 2008]. A coarse search for the optimal *mtry* was performed in the current study using *mtry* values of 1, $2\sqrt{p}$, 0.1p, 0.5p and p. The parameterization that produced the lowest final OOB-ER was chosen as the optimal *mtry*.

Number of Trees to Grow (*ntree*)

Another important consideration is how many trees to grow. This is also a dataset dependent factor, where stronger predictors lead to quicker convergence. While for

prediction purposes few trees are often necessary, and the OOB-ER will generally converge rapidly, for VI, more trees will generally lead to refinement and stability in VI [Genuer et al., 2008].

The main trade-off with growing a larger number of trees is the computation cost required. In the current study, trees were grown until the OOB-ER stabilized. Additional trees were then grown to ensure stability.

Weighting

The final tuning parameter, which was not considered in this analysis, is weighting. In classification, with uneven classes, an unweighted classification scheme will be biased towards the majority class. The typical strategy is to re-weight the classes so that they are balanced, the practice used within the Salford Systems implementation of RF, and the default in the R implementation. Unfortunately, class weighting cannot be altered in the Salford Systems version, so it could not be tested as a tuning parameter. However, internal testing on a more flexible version of RF showed no added benefit to changing the weighting.

2.2.4 Data Configurations

Sparsity Pruning

As noted, it is expected that the vast majority of SNPs in a GWA study do not impact risk for disease, and therefore, are simply noise. The goal of any algorithm should be to separate noise from signal, providing a sparse solution. A sparse solution is indicated when the VI is either 0 or negative. Such a VI indicates that the variable was either never selected into a tree, or when it was selected, permutation did not increase the prediction error. Sparse solutions provide a convenient way to remove unimportant data from the analysis.

Sparsity is a function of both *mtry* and *ntree*, with a higher *mtry* leading to greater sparsity and a higher *ntree* leading to less sparsity. One proposed strategy is to sequentially remove genes by dropping the bottom 20% or 50%, and perform successive runs until there is a noticeable increase in prediction error [Díaz-Uriarte and Alvarez de Andrés, 2006]. Utilizing the natural sparseness in the dataset, the results of each RF run were examined and sparse SNPs were dropped. The RF analysis was then re-run until prediction error stabilized. While this will give a biased estimate of the prediction error for the model [Svetnik et al., 2004], it can still be used to judge model quality. This sub-sampling process was repeated in the current study until the final OOB error-rate stabilized or increased.

Removing Strong Associations

RF searches over multiple variables finding solutions based on joint and conditional effects. Since VI score is dependent on where a variable lies in the tree, it is possible that variable with strong effects may mask weaker, yet important effects by pushing them down the tree. It is well established that the HLA region within the major histocompatibility complex (MHC) on chromosome 6p is strongly associated with MS [Oksenberg and Barcellos, 2005]. Therefore, to search for weaker non-MHC effects, RF analysis was performed in the current study after removing chromosome 6p marker data.

Linkage Disequilibrium

An important consideration when applying RF to GWA data is the large degree of LD among SNPs. VI is calculated from the number of trees in which a variable appears. Therefore, two SNPs that are in perfect LD will appear in trees about half as often as each individual one may appear by itself, effectively lowering the VI of each SNP. While this does not present a problem for prediction, it can skew the VI rankings [Genuer et al., 2008]. Two proposed solutions have been to calculate VI independently of the number of trees in which the variable appears [Meng et al., 2009] or as conditional on other variables in the tree [Strobl et al., 2007].

PLINK [Purcell et al., 2007] provides two methods of LD pruning based on r^2 and R^2. r^2 is a traditional pairwise LD measure, though not based on phased haplotypes. R^2 is the multiple correlation coefficient based on a sliding window. Using PLINK, SNPs with a multiple correlation coefficient (R^2) of 0.99, 0.90, 0.80, 0.50 and 0.33 were removed from the MS case-control dataset for comparison. This resulted in pruning between 22% and 76% of the original data which had the side benefit of increasing computational efficiency. While this does not necessarily aid in determining the causal SNP (that one may be pruned out) it does improve detecting a region of interest.

2.2.5 Reliability of Results Obtained from RF

Since RF is a Monte-Carlo process, random variation may influence VI results, particularly if enough trees are not grown. While, work has indicated that RF results are relatively stable [Genuer et al., 2008] and our own internal testing has confirmed this, it is important to grow large forests and do multiple runs when possible. Reliability of final RF results was examined by re-running RF with the final dataset configuration, parameterization and sub-sampling process, changing just the seed in the random number generator. While more than one re-run would be ideal, the VI measures are unlikely to be unstable given that two runs were performed.

2.2.6 Comparison of RF Results to Original GWA study

The original MS GWA study identified, with replication, 16 SNPs across 13 genes as associated with MS [Hafler et al., 2007]. An important consideration for the current study was whether RF could identify additional genes of interest, as well as duplicate the original findings based on univariate testing. Duplication was considered present when a SNP identified by RF was: (1) among the original 16 SNPs, or (2) a SNP that was tagged by one of the 16 SNPs identified in the original GWA study. PLINK was used to identify tagged SNPs using an r^2 threshold of 0.5.

2.2.7 Analysis Strategy

Figure 2.2 presents the analysis plan. The primary method for choosing tuning parameters was minimization of the OOB-ER, as this is the best indication of model quality. Determining how many results to report is more subjective since the VI measure does not constitute a formal hypothesis test. To help guide interpretation of RF results for follow-up, we plotted VI scores. A sloping line with an elbow (Figure 2.3 was observed most often around the top 25, so this was chosen as the cutoff for an important result. While this is an inherently adhoc solution to determining appropriate cutoffs, to this point solid statistical properties of VI scores have not been determined to aide in a more objective approach.

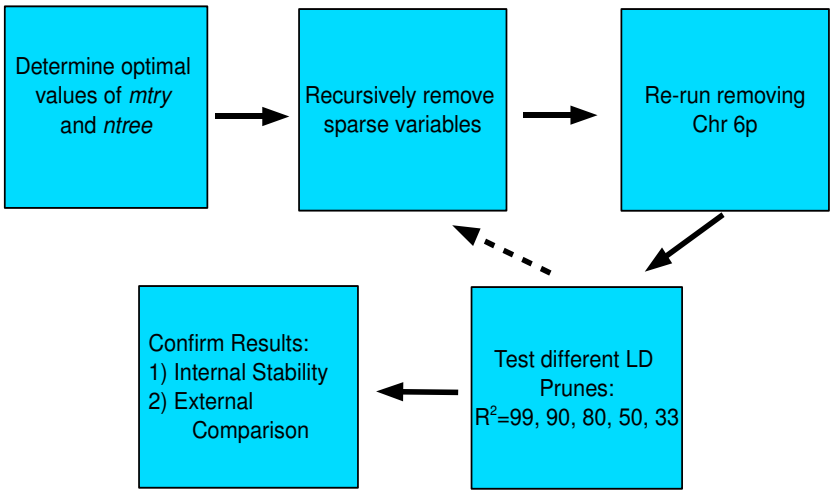


Figure 2.2: Flow Plan for RF analysis. The full MS case-control dataset was analyzed, searching for the optimal *mtry* & *ntree*, along with sparsity pruning, as necessary. Two runs were then conducted, one without any 6p genotypes, and one with data for a single 6p SNP. Finally, LD pruning was explored. After the best data configuration was found, RF analysis was re-run to examine stability of results. The final RF results were compared to the original GWA results [Hafler et al., 2007].

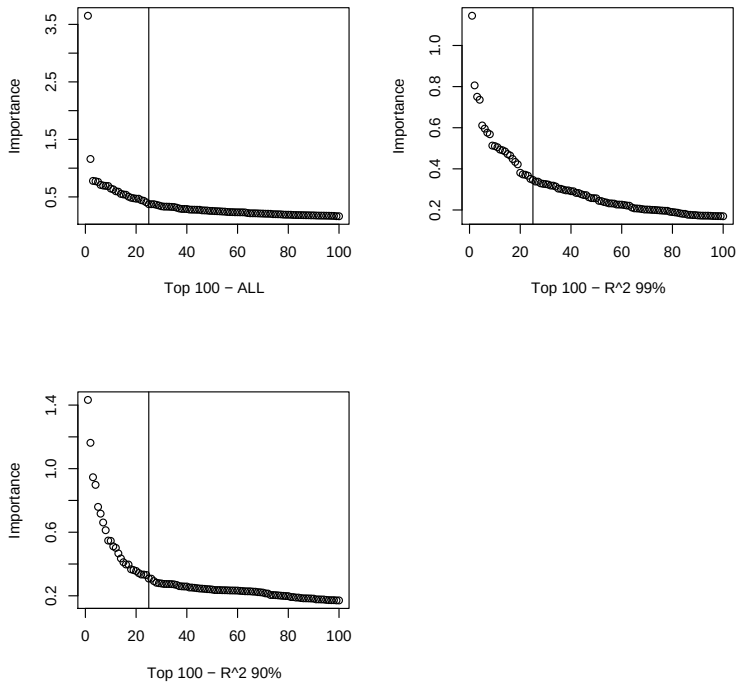


Figure 2.3: The three plots represent the VI measures for the full dataset with chromosome 6p data removed, the $R^2 = 0.99$ run and the $R^2 = 0.90$ run. An elbow is present in all three plots around 25 markers (designated with the vertical line).

2.3 Results

2.3.1 Tuning Parameters

Number of variables to choose per node (*mtry*)

The first parameter considered was *mtry* since this has the greatest impact on the OOB-ER. Figure 2.4 shows the OOB-ER for different values of *mtry*. The typically suggested value of *mtry* of around $2\sqrt{p}$ is not sufficient for GWA data, as the OOB-ER is minimized with an *mtry* around .1p. Among the higher *mtry* values, there was little distinction between them with regard to OOB-ER. The top SNPs from the *mtry* of .1p are shown in Table 2.1.

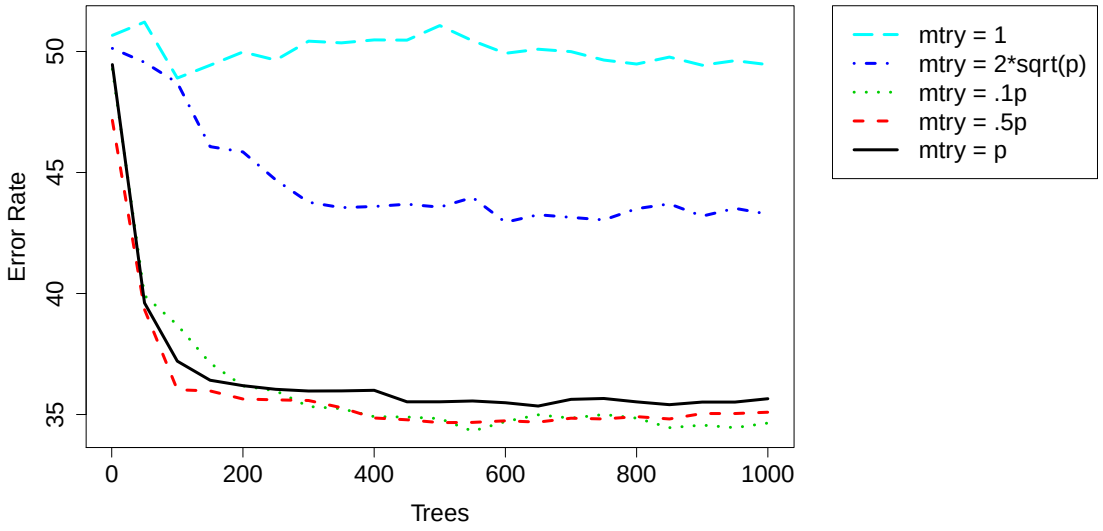


Figure 2.4: An examination of the error-rate across different *mtrys*. The larger *mtrys* of .1p and above clearly lead to a much lower error rate than the more traditional lower values. .1p seems to minimize the overall OOB error-rate though not by much. Convergence seems to occur around 200 400 trees.

Another consideration is the sparsity induced by the *mtry* factor. As expected, sparsity increases with *mtry*, though this is most dramatic after increasing to *mtry* = p (Figure 2.5).

Number of Trees to Grow (*ntree*)

Using *mtry* = .1p, forests of size 50, 250, 500, 1,000, 1,500 and 2,000 trees were grown. It is clear that the OOB-ER leveled off around 250 trees (see Figure 4) and

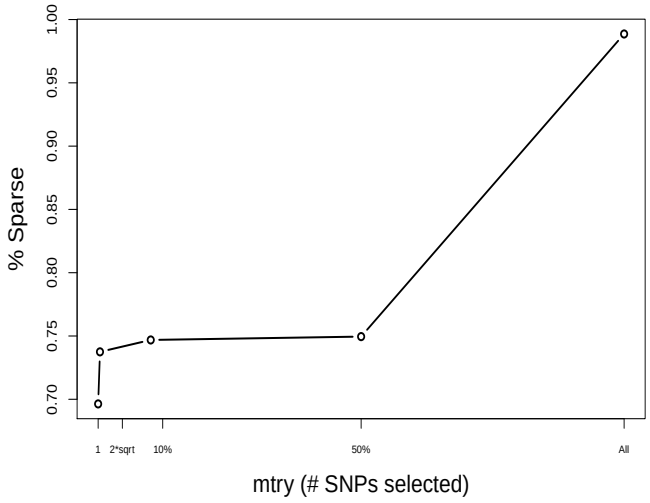


Figure 2.5: As expected, sparsity increases as a function of *mtry*. There is the most dramatic increase after moving from an *mtry* of .5p to p.

1,000 trees was used as a reliable forest size. However, for datasets without chromosome 6p and only weak predictors (see below), it took more than 4,000 and sometimes 8,000 trees for convergence. In those cases, 5,000 and 10,000 trees, respectively, were grown. More trees led to a less sparse result, as expected, with nearly a linear decrease through 2,000 trees.

Top SNPs identified by Random Forests in MS case-control dataset

Chr	SNP	Gene	MAF	Rank	CHISQ	P-Value
6	rs3129900	*C6orf10*	0.17	1	272.2	3.75×10^{-61}
6	rs3129934	*C6orf10*	0.17	2	274.4	1.28×10^{-61}
6	rs9270986	*HLA Tag SNP*	0.17	3	274.6	1.14×10^{-61}
6	rs3129768	*HLA-DQA* (70bp)*	0.20	4	238.9	3.14×10^{-53}
6	rs2647046	*HLA-DQA2* (8.5kb)*	0.39	5	113.9	1.38×10^{-26}
6	rs3129932	*C6orf10*	0.23	6	219.8	1.02×10^{-49}
6	rs9275572	*HLA-DQA2* (2.1kb)*	0.42	7	101.5	7.24×10^{-24}
6	rs3131294	*NOTCH4*	0.14	8	215.4	9.26×10^{-49}
6	rs910049	*C6orf10*	0.24	9	222.2	2.98×10^{-50}
6	rs2894249	*C6orf10*	0.23	10	220.7	6.28×10^{-50}
6	rs3135377	*HLA-DRA* (80.6kb)*	0.21	11	217.9	2.60×10^{-49}
6	rs9469220	*HLA-DQA2* (18.5kb)*	0.50	12	99.2	2.28×10^{-23}
6	rs7194	*HLA-DRA*	0.40	13	129.7	4.69×10^{-30}
6	rs6457620	*HLA-DQB1* (137.5kb)*	0.49	14	96.03	1.13×10^{-22}
6	rs3130287	*TNXB*	0.15	15	181.2	2.72×10^{-41}
6	rs6457617	*HLA-DQB1 (137.4kb)*	0.49	16	96.03	1.13×10^{-22}
6	rs6936204	*C6orf10* (14.6kb)*	0.36	17	113.3	1.83×10^{-26}
12	rs1805755	*M6PR*	¡.01	18	73.42	1.05×10^{-17}
12	rs1716167	*MPHOSPH9*	0.21	19	22.38	2.23×10^{-6}
7	rs17708673	*C7orf25 (106.2kb)*	0.16	20	6.357	1.17×10^{-2}
6	rs9268877	*HLA-DRA* (126.3kb)*	0.42	21	74.57	5.85×10^{-18}
6	rs9276440	*HLA-DQA2*	0.45	22	83.75	5.63×10^{-20}
6	rs2621383	*HLA-DOB* (825.5kb)*	0.37	23	82.72	9.44×10^{-20}
22	rs80515	*FAM19A5* (1.4mb)*	0.10	24	3.751	5.28×10^{-2}
20	rs2425754	*CDH22* (580.3 kb)*	0.15	25	4.193	4.06×10^{-2}

Table 2.1: The top 25 SNPs from RF analysis of the whole dataset are shown above. Most of the top SNPs are on chromosome 6p within the HLA region. The minor allele frequency (MAF) is derived from controls and the χ^2-statistic is from univariate testing. *Indicates that the gene is the closest gene with distance.

2.3.2 Data Configurations

Sparsity Pruning

When using the full dataset for RF analysis, SNPs within the HLA region of chromosome 6p were consistently selected as the most important variables (Table 2.1). This is not surprising, as some SNPs in that region had a marginal χ^2-statistic as large as 274. The final error rate of 35% is identical to a simple classification based just on genotypes for the three most highly associated SNPs (rs3129900, rs3129934, rs9370986). RF results based on analysis of all SNPs from chromosome 6p resulted in the same 35% error rate.

Removing Chromosome 6p

After removing all SNPs on chromosome 6p (p = 8,335), the initial run of 317,472 SNPs produced an error-rate of 48% after 1,000 trees, and using both an *mtry* of .1p and p. The resulting forest based on *mtry* of .1p was 74% sparse (82,527 SNPs retained). Using *mtry* = p, the forest was 99% sparse (4,219 SNPs retained).

For the *mtry* = p run, re-running RF analysis with the reduced dataset produced an error-rate of 26%, and required approximately 4,000 trees to converge. Repeating this sub-sampling process two more times produced an error-rate of 21%. After a fourth run, the OOB error-rate remained at 21%, suggesting that three sub-samples were sufficient. For the 10% run, the final OOB error-rate was 37% and contained 25,000 SNPs.

Overall, results suggest there is predictive structure (differences between MS cases and controls) beyond chromosome 6p, and that aggressive pruning of the initial *mtry*= p is more effective for discovering that structure. The top 25 SNPs derived from RF analysis without chromosome 6p markers are shown in Table 2.2.

Linkage Disequilibrium

The final consideration was the effect of pruning SNPs based on LD. The dataset without any markers for chromosome 6p was used and the same sub-sampling strategy was followed. Figure 2.6 shows final error-rates for the six LD configurations investigated, along with the full dataset. The number of SNPs in each configuration is included.

While pruning past an R^2 of 0.90 (LD90) results in a higher final error-rate and suggests a loss of information, it is hard to determine which approach is best when solutions based on full data, LD99 and LD90 are compared. Examination of the top 25 SNPs from the three configurations (full, LD99 and LD90; Tables 2.2 - 2.4), reveals that most of the SNPs were located within a gene (14, 14, and 15 respectively). However, the LD90 solution identified SNPs within more unique genes (14) compared to the other configurations (9 and 11). In addition to identification of potentially

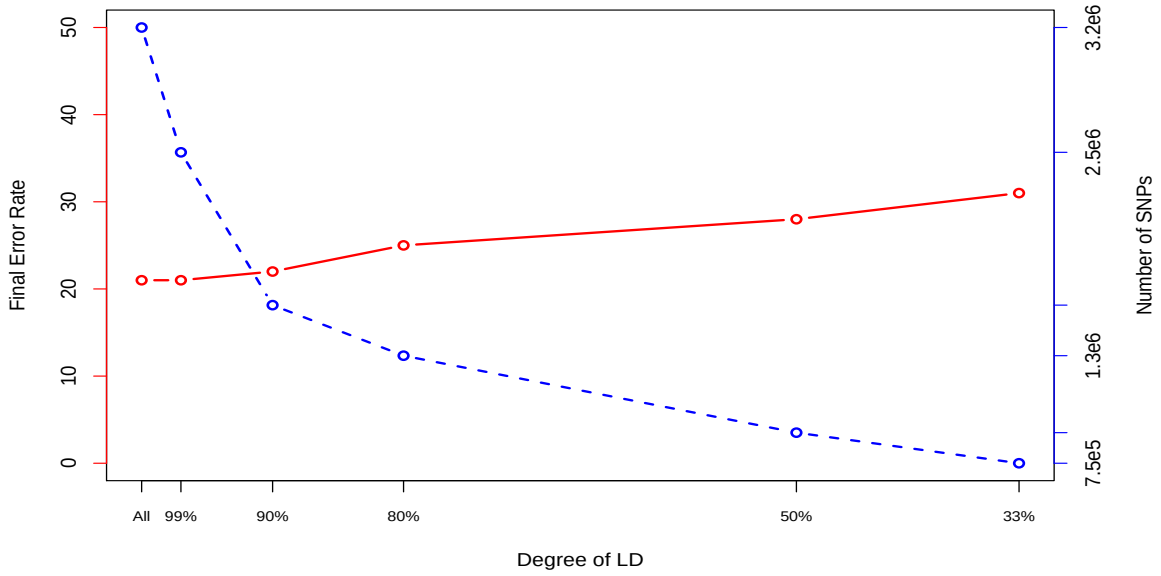


Figure 2.6: In the red line we see the OOB error rate across the different LD prunes. There is little information lost going from the full data to pruning at 99% and even 90%. Thereafter there is more loss of information. The blue line shows the number of SNPs that were in each RF analysis.

functional SNPs, the majority of top results show strong marginal associations ($p \simeq 10^{-5}$) but do not meet established criteria for genome-wide significance [Pearson and Manolio, 2008]. When the top 25 SNP results from each configuration were compared, both overlapping and unique genes are observed. Genes not previously associated with MS were among the top hits, specifically *CTNNA3*, *MPHOSPH9*, *PHACTR2*, and *IL7*.

Top SNPs identified by Random Forests in MS case-control dataset without 6p data

Chr	SNP	Gene	MAF	Rank	CHISQ	P-Value
12	rs1805755	*M6PR*	< 0.01	1	73.42	1.05×10^{-17}
7	rs6467970	*SEMA3A* (44.1 kb)*	0.19	2	19.71	9.00×10^{-6}
10	rs10823051	*CTNNA3*	0.16	3	17.61	2.71×10^{-5}
1	rs10754012	*RGS1* (3.3 mb)*	0.23	4	22.24	2.41×10^{-5}
12	rs1716167	*MPHOSPH9*	0.21	5	22.38	2.23×10^{-6}
6	rs1015340	*PHACTR2*	0.47	6	14.86	1.16×10^{-4}
10	rs7068990	*PPAPDC1A* (137.6 kb)*	0.23	7	17.65	2.65×10^{-5}
8	rs1466526	*FAM164A* (86.0 kb)*	0.25	8	15.60	7.84×10^{-5}
6	rs1040638	*PHACTR2*	0.48	9	13.82	2.01×10^{-4}
7	rs16217	*NPY* (292.9 kb)*	0.26	10	7.14	7.53×10^{-3}
12	rs1106240	*PITPNM2*	0.20	11	18.71	1.52×10^{-5}
8	rs4739135	*FAM164A* (98.8 kb)*	0.19	12	16.33	5.34×10^{-5}
18	rs4798684	*ADCYAP1* (19.8 kb)*	0.30	13	13.41	2.51×10^{-4}
6	rs1015341	*PHACTR2*	0.47	14	14.88	1.14×10^{-4}
12	rs2695478	*MPHOSPH9*	0.20	15	17.30	3.19×10^{-5}
1	rs11800848	*EVI5*	0.26	16	19.04	1.28×10^{-5}
9	rs6993386	*IL7*	0.32	17	17.95	2.27×10^{-5}
6	rs6915752	*PHACTR2*	0.45	18	17.71	2.57×10^{-5}
20	rs2223712	*BTBD3* (3.6 kb)*	0.19	19	11.86	5.73×10^{-4}
10	rs7092549	*PPAPDC1A* (140.0 kb)*	0.23	20	17.06	3.62×10^{-5}
6	rs9376783	*PHACTR2*	0.45	21	17.26	3.27×10^{-5}
17	rs17652139	*CCL2* (3.0 mb)*	0.23	22	11.81	5.88×10^{-4}
7	rs740295	*MGC87402*	0.31	23	7.89	4.96×10^{-3}
5	rs156823	*ARL15*	0.47	24	12.13	4.97×10^{-4}
18	rs7241142	*ADCYAP1* (20.7 kb)*	0.30	25	10.81	1.01×10^{-3}

Table 2.2: The top 25 SNPs from RF analysis of the dataset without chromosome 6 SNPs are shown above. The minor allele frequency (MAF) is derived from controls and the χ^2-statistic is from univariate testing. *Indicates that the gene is the closest gene with distance.

Top RF SNPs in the MS case-control dataset with LD pruning $R^2 = 0.99$

Chr	SNP	Gene	MAF	Rank	CHISQ	P-Value
7	rs6467970	*SEMA3A* (44.1 kb)*	0.19	1	19.71	9.00×10^{-6}
8	rs1466526	*FAM164A* (86.0 kb)*	0.25	2	15.60	7.84×10^{-5}
12	rs1716167	*MPHOSPH9*	0.21	3	22.38	2.23×10^{-6}
10	rs10823051	*CTNNA3*	0.16	4	17.61	2.71×10^{-5}
1	rs11800848	*EVI5*	0.26	5	19.04	1.28×10^{-5}
17	rs17652139	*CCL2* (3.0 mb)*	0.23	6	11.81	5.88×10^{-4}
6	rs1015341	*PHACTR2*	0.47	7	14.88	1.14×10^{-4}
7	rs16217	*NPY* (292.9 kb)*	0.26	8	7.14	7.53×10^{-3}
1	rs12743520	*EVI5*	0.26	9	18.61	1.61×10^{-5}
6	rs1040638	*PHACTR2*	0.48	10	13.82	2.01×10^{-4}
18	rs4798684	*ADYCAP1* (19.8 kb)*	0.30	11	13.41	2.51×10^{-4}
8	rs6993386	*IL7*	0.32	12	17.95	2.27×10^{-5}
1	rs2760524	*RGS1* (3.3 mb)*	0.19	13	20.00	7.76×10^{-6}
10	rs7092549	*PPAPDC1A* (140.0 kb)*	0.23	14	17.06	3.62×10^{-5}
20	rs2223712	*BTBD3* (3.6 kb)*	0.19	15	11.86	5.72×10^{-5}
7	rs740295	*MGC87402*	0.31	16	7.90	4.96×10^{-5}
1	rs282177	*RPS6KA1*	0.26	17	17.01	3.72×10^{-5}
6	rs6570578	*PHACTR2*	0.45	18	17.00	3.72×10^{-5}
10	rs7068990	*PPAPDC1A* (137.6 kb)*	0.23	19	17.65	2.65×10^{-5}
1	rs1359062	*RGS1* (3.3 mb)*	0.19	20	18.49	1.71×10^{-5}
2	rs698853	*LOC100302652*	0.28	21	16.75	4.26×10^{-5}
7	rs156293	*NPY* (313.8 kb)*	0.22	22	9.56	1.99×10^{-5}
16	rs6499946	*KLKBL4*	0.22	23	12.83	3.41×10^{-4}
2	rs7583622	*ASB3*	0.23	24	17.07	3.60×10^{-5}
20	rs17408919	*PAK7*	0.23	25	11.63	6.50×10^{-4}

Table 2.3: The top 25 SNPs from RF analysis of the dataset without chromosome 6 SNPs are shown above in the $R^2 = 0.99$ runs after 3 sub-samplings. Results are similar to analysis of full dataset. *Indicates that the gene is the closest gene with distance.

Top RF SNPs in the MS case-control dataset with LD pruning $R^2 = 0.90$

Chr	SNP	Gene	MAF	Rank	CHISQ	P-Value
7	rs6467970	*SEMA3A* (44.1 kb)*	0.19	1	19.71	9.00×10^{-6}
8	rs1466526	*FAM164A* (86.0 kb)*	0.25	2	15.60	7.84×10^{-5}
10	rs10823051	*CTNNA3*	0.16	3	17.61	2.71×10^{-5}
1	rs10754012	*RGS1* (3.3 mb)*	0.23	4	22.24	2.41×10^{-6}
6	rs1040638	*PHACTR2*	0.48	5	13.82	2.01×10^{-4}
8	rs4739135	*FAM164A* (98.8 kb)*	0.19	6	16.33	5.34×10^{-5}
18	rs4798684	*ADCYAP1* (19.8 kb)*	0.30	7	13.41	2.51×10^{-4}
10	rs7092549	*PPAPDC1A* (140.0 kb)*	0.23	8	17.06	3.62×10^{-5}
14	rs10483442	*NPAS3*	0.19	9	13.88	1.95×10^{-4}
20	rs2223712	*BTBD3* (3.6 kb)*	0.19	10	11.86	2.71×10^{-4}
12	rs1106240	*PITPNM2*	0.20	11	18.71	1.52×10^{-5}
5	rs156823	*ARL15*	0.47	12	12.13	4.97×10^{-4}
9	rs10975130	*KANK1*	0.16	13	24.04	7.81×10^{-6}
12	rs12578774	*AACS* (1.3 mb)*	0.31	14	19.67	9.19×10^{-6}
2	rs7583622	*ASB3*	0.23	15	17.07	3.60×10^{-5}
6	rs6570578	*PHACTR2*	0.45	16	17.00	3.73×10^{-5}
1	rs282177	*RPS6KA1*	0.26	17	17.01	3.72×10^{-5}
5	rs11949767	*MXD3* (59.7 kb)*	0.26	18	17.79	2.47×10^{-5}
10	rs7427	*MSRB2*	0.36	19	10.64	1.10×10^{-3}
8	rs1879818	*TRAPPC9*	0.30	20	7.35	6.72×10^{-3}
16	rs1974876	*CCDC113*	0.15	21	10.91	9.55×10^{-3}
17	rs11651517	*GAS7*	0.43	22	10.29	1.34×10^{-3}
20	rs6018946	*BLCAP* (581.3 kb)*	0.34	23	16.85	4.04×10^{-5}
2	rs11694785	*ARHGAP25*	0.40	24	10.15	1.44×10^{-3}
2	rs6746541	*ATOH8*	0.35	25	16.06	6.14×10^{-5}

Table 2.4: The top 25 SNPs from RF analysis of the dataset without chromosome 6 SNPs are shown above in the $R^2 = 0.90$ runs after 3 sub-samplings. Results are similar to the analysis of full dataset, though there is more heterogeneity in the top findings, owing primarily to LD pruning. *Indicates that the gene is the closest gene with distance.

2.3.3 Reliability of Results Obtained from RF

The final three data configurations (full data, LD99 and LD90) were re-analyzed, changing only the random number seed. For all three configurations, at least 19 of the top 25 SNPs were in the final results after sparsity pruning. This suggests that even after changing the seed, RF results are very stable.

Comparison of Results to Original GWA Study

Finally, the RF results were compared with replicated results from the original MS GWA study [Hafler et al., 2007]. In all, 4 of 13 MS genes were directly identified by one of the three data configurations. The strongest evidence came from SNPs in *EVI5* and *KANK1* with a suggestion of duplication in *IL2RA*.

2.4 Discussion

This study is the first application of RF, and one of few machine learning applications, to the analysis of a GWA dataset. The goals were to outline methodological considerations for applying RF to large GWA data, and to identify potential novel MS associations. Given what is currently known about the genetics of MS, it was not surprising that a strong classifier could be constructed by RF based on data for multiple SNPs within the MHC. Among the strongest effects (most important SNP predictors of MS as outcome) was rs9271366, which has been previously shown to tag DRB1*1501 with $r^2 = 0.98$ [Australia and Consortium, 2009]. Interestingly, once the 6p effect was removed from analyses, a strong classifier based on non-MHC data emerged. Results suggest that sparsity pruning provides a means to discover new associations with RF, although the final error-rate is biased [Svetnik et al., 2004].

RF analyses consistently identified four non-MHC genes as important to distinguishing MS cases from controls. These were: *MPHOSPH9*, *CTNNA3*, *PHACTR2* and *IL7*. *MPHOSPH9* up-regulates neuronal functioning [Ward et al., 2007], and interestingly, variation within this locus has recently shown suggestive evidence for association in a much larger meta-analysis that included 2,624 MS cases and 7,220 controls [deJager et al., 2009]. *CTNNA3* is a cell adhesion gene that has been associated with Alzheimer's disease [Morgan et al., 2008]. *PHACTR2* is involved in phosphate and actin regulation and has been implicated in Parkinson's disease [Wider et al., 2009]. Finally, *IL7* is an important immune system gene involved in T and B cell production and has been implicated in other autoimmune diseases, notably rheumatoid arthritis, but not MS [van Roon and Lafeber, 2008]. It is important to note that although SNPs within *CTNNA3*, *MPHOSPH9*, *PHACTR2* and *IL7* were among the top RF results, associations for these SNPs based on univariate analyses would not meet criteria for genome-wide significance Pearson and Manolio [2008] (Tables 2.2 - 2.4). As a point of comparison, statistical power based on univariate testing was high

in our dataset (n=931 cases and 2,431 controls) for detecting an effect size per allele (or allelic odds ratio) of 1.5 (assuming MAF=0.15-0.50 and $\alpha = 1.5 * 10^{-7}$), However, power was quite limited to detect smaller effect sizes, for example, 1.3 or 1.2, where $\sim 5 - 30\%$ and $\sim 0.5 - 3\%$ power, respectively, was present. To date, replicated non-MHC MS genes have demonstrated very modest effects of 1.2 or even smaller [Hafler et al., 2007, deJager et al., 2009, Australia and Consortium, 2009]. New results from the current study will require further replication in a larger, independent dataset, but underscore the utility of using more than one analytical method to identify genetic associations.

RF results were also compared to findings from the original MS case-control study using the same dataset, with duplication defined as a either the original SNP or one tagged by that SNP among the top RF results. Two previously reported genes, *EVI5* and *KANK1*, were among the top RF findings in the current study. There was also a suggestion of importance based on RF analyses for *IL2RA* and perhaps *CBLB*.

Methodologically, it was shown that RF can be applied to large GWA datasets, but certain standard assumptions cannot always be made. The OOB-ER was relied upon to guide decision making about tuning parameters and data configuration. Even though the focus of the current study was not prediction, this error-rate is valuable for determining the quality of RF results. First, when working with large, sparse data, the default value of *mtry* needs to be increased in order to improve learning. Even for the sub-sampled data sets, generally an *mtry* = .1p was the optimal setting. It was also found that the number of trees necessary to reach stability depended greatly on the strength of the inputs. For the data configurations with chromosome 6p genotype data, stability was reached within 250 trees, while for the data configurations without chromosome 6p data, stability was often not reached until at least 4,000 trees were generated. LD pruning can be an effective means of reducing data size without significant loss of information. Also removing sparse variables proved to be highly effective and resulted in much more efficient learning. It was established that some very strong effects (chromosome 6p) can mask weaker, yet potentially interesting effects. Prediction based on genetic data that did not include HLA region SNPs was surprisingly strong. Finally, one needs to consider the coding of the allelic data. Coding the data on a dosage scale allows for a flexible examination of genetic effects. Upon settling on a final configuration(s), doing multiple runs of RF is necessary to examine the reliability of the VI measures.

More work is needed to achieve a better understanding of the RF algorithm and how best to apply it to large GWA datasets. The theoretical basis for RF as a predictor is well understood, but less is known about VI. Unlike p-values, there is no strict criterion for distinguishing between important and non-important variables. Our decision to focus on the top 25 results was based on graphing results, and in that sense was fairly qualitative. Ideally, one would use permutation to assess the significance of the VI measures, however this is not feasible with these large datasets. Work is ongoing to determine valid cutoffs for VI measures. Also, only one form of VI was used

in this analysis (permutation), but another general VI exists for classification based on the Gini criterion (the optimizing criteria used to construct the tree). Work is also ongoing to define more targeted measures of VI, particularly for SNP data. Furthermore, as discussed, LD between SNPs and other correlated data are problematic for RF due to the way VI is calculated and we are currently exploring alternative VI calculations. Finally, further work is also needed to leverage additional information from the forest of trees. Little work has been done on clustering observations in RF. The tree structure can also be used for identifying extensive regions of interactions and genetic networks and predictors important to specific disease phenotypes.

2.5 Conclusions

This study represents one of the first successful applications of a machine learning algorithm to GWA data. Machine learning algorithms require fewer assumptions about the data generating distribution, and therefore, offer a very flexible approach to data analysis. Our results show the RF algorithm is both computationally feasible and sensible for analyses of large GWA datasets. Computation time ranged from a few minutes to a few days depending on the number of variables. Our results support findings from previous genetic studies in MS, and more importantly, new candidates emerged that strongly warrant further investigation.

A unique approach to analyzing complex genetic data is described in the current study. As other machine learning algorithms are expanded to accommodate large GWA datasets, one can apply an array of algorithms to a large dataset, and then aggregate results across methods to determine which markers or genes may be of greatest interest for future studies. Such ensemble learners are common in the machine learning literature [Hastie et al., 2009], and are becoming more applicable to larger genetic datasets [van der Laan et al., 2007].

Bibliography

Salford systems. URL `http://salford-systems.com/`.

Australia and New Zealand Multiple Sclerosis Genetics Consortium. Genome-wide association study identifies new multiple sclerosis susceptibility loci on chromosomes 12 and 20. *Nature Genetics*, 41:824–828, 2009.

O. Bembom, M. L. Petersen, S. Rhee, W. J. Fessel, S. E. Sinisi, R. W. Shafer, and M. J. van der Laan. Biomarker discovery using targeted maximum likelihood estimation: Application to the treatment of antiretroviral resistant hiv infection. Technical Report 221, U.C. Berkeley Division of Biostatistics Working Paper Series, August 2007.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996a.

L. Breiman. Out-of-bag estimation. Technical report, UC Berkeley, 1996b.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

S. R. Browning and B. L. Browning. Rapid and accurate haplotype phasing and missing-data inference for whole-genome association studies by use of localized haplotype clustering. *American Journal Human Genetics*, 81:1084–1097, Nov 2007.

A. Bureau, J. Dupuis, K. Falls, K. L. Lunetta, B. Hayward, T. P. Keith, and P. Van Eerdewegh. Identifying SNPs predictive of phenotype using random forests. *Genetic Epidemiology*, 28:171–182, Feb 2005.

P L deJager, X Jia, J Wang, P I Q deBakker, L Ottoboni, N T Aggarwal, L Piccio, S Raychaudhuri, D Tran, C Aubin, R Briskin, S Romano, and IMSGC. Meta-analysis of genome scans and replication identify CD6, IRF8 and TNFRSF1A as new multiple sclerosis susceptibility loci. *Nature Genetics*, 41:776–782, July 2009.

R. Díaz-Uriarte and S. Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.

R. Genuer, J.M. Poggi, and C. Tuleau. Random Forests: some methodological insights. Technical report, INRIA, 2008. URL `http://hal.inria.fr/inria-00340725/en/`.

B. Glaser, I. Nikolov, D. Chubb, M. L. Hamshere, R. Segurado, V. Moskvina, and P. Holmans. Analyses of single marker and pairwise effects of candidate loci for rheumatoid arthritis using logistic regression and random forests. *BMC Proceedings*, 1 Suppl 1:S54, 2007.

D. A. Hafler, A. Compston, S. Sawcer, E. S. Lander, M. J. Daly, P. L. De Jager, P. I. de Bakker, S. B. Gabriel, D. B. Mirel, A. J. Ivinson, M. A. Pericak-Vance, S. G. Gregory, J. D. Rioux, J. L. McCauley, J. L. Haines, L. F. Barcellos, B. Cree, J. R. Oksenberg, and S. L. Hauser. Risk alleles for multiple sclerosis identified by a genomewide study. *New England Journal of Medicine*, 357:851–862, Aug 2007.

T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning*. Springer, New York, 2 edition, 2009.

A. G. Heidema, J. M. Boer, N. Nagelkerke, E. C. Mariman, D. L. van der A, and E. J. Feskens. The challenge for genetic epidemiologists: how to analyze large numbers of SNPs in relation to complex diseases. *BMC Genetics*, 7:23, 2006.

C. Kooperberg and I. Ruczinski. Identifying interacting SNPs using Monte Carlo logic regression. *Genetic Epidemiology*, 28:157–170, Feb 2005.

A. Liaw and M. Wiener. Classification and regression by randomforest. *Rnews*, 2: 18–22, 2002.

K. L. Lunetta, L. B. Hayward, J. Segal, and P. Van Eerdewegh. Screening large-scale association study data: exploiting interactions using random forests. *BMC Genetics*, 5:32, 2004.

Y. A. Meng, Y. Yu, L. A. Cupples, L. A. Farrer, and K. L. Lunetta. Performance of random forest when SNPs are in linkage disequilibrium. *BMC Bioinformatics*, 10: 78, 2009.

A. R. Morgan, G. Hamilton, D. Turic, L. Jehu, D. Harold, R. Abraham, P. Hollingworth, V. Moskvina, C. Brayne, D. C. Rubinsztein, A. Lynch, B. Lawlor, M. Gill, M. O'Donovan, J. Powell, S. Lovestone, J. Williams, and M. J. Owen. Association analysis of 528 intra-genic SNPs in a region of chromosome 10 linked to late onset Alzheimer's disease. *Am. J. Med. Genet. B Neuropsychiatr. Genet.*, 147B:727–731, Sep 2008.

A. A. Motsinger and M. D. Ritchie. Multifactor dimensionality reduction: an analysis strategy for modelling and detecting gene-gene interactions in human genetics and pharmacogenomics studies. *Human Genomics*, 2:318–328, Mar 2006.

BA Nonyane and AS Foulkes. Application of two machine learning algorithms to genetic association studies in the presence of covariates. *BMC Genetics*, 9:71, Nov 2008.

J. R. Oksenberg and L. F. Barcellos. Multiple sclerosis genetics: leaving no stone unturned. *Genes and Immunity*, 6:375–387, Aug 2005.

T A Pearson and T A Manolio. How to interpret a genome-wide association study. *JAMA*, 299:1335–1344, 2008.

S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. de Bakker, M. J. Daly, and P. C. Sham. PLINK: a tool set for whole-genome association and population-based linkage analyses. *American Journal Human Genetics*, 81:559–575, Sep 2007.

C. Strobl, A. L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics*, 8:25, 2007.

Y. V. Sun, Z. Cai, K. Desai, R. Lawrance, R. Leff, A. Jawaid, S. L. Kardia, and H. Yang. Classification of rheumatoid arthritis status with candidate gene and genome-wide single-nucleotide polymorphisms using random forests. *BMC Proceedings*, 1 Suppl 1:S62, 2007.

V. Svetnik, A. Liaw, and C. Tong. Variable selection in random forest with application to quantitative structureactivity relationship. In N. Intrator and F. Masulli, editors, *Proceedings of the 7th Course on Ensemble Methods for Learning Machines*. Springer-Verlag, 2004.

M. J. van der Laan, E. C. Polley, and A. E. Hubbard. Superlearner. *Statistical Applications in Genetics & Molecular Biology*, 6, 2007.

J. A. van Roon and F. P. Lafeber. Role of interleukin-7 in degenerative and inflammatory joint diseases. *Arthritis Research & Therapy*, 10:107, 2008.

G. R. Ward, S. O. Franklin, T. M. Gerald, K. T. Dempsey, D. E. Clodfelter, D. J. Krissinger, K. M. Patel, K. E. Vrana, and A. C. Howlett. Glucocorticoids plus opioids up-regulate genes that influence neuronal function. *Cellular and Molecular Neurobiology*, 27:651–660, Aug 2007.

C. Wider, S. J. Lincoln, M. G. Heckman, N. N. Diehl, J. T. Stone, K. Haugarvoll, J. O. Aasly, J. M. Gibson, T. Lynch, A. Rajput, M. L. Rajput, R. J. Uitti, Z. K. Wszolek, M. J. Farrer, and O. A. Ross. Phactr2 and Parkinson's disease. *Neuroscience Letters*, 453:9–11, Mar 2009.

WTCCC. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447:661–678, Jun 2007.

Y Yoon, J Song, SH Hong, and JQ Kim. Analysis of multiple single nucleotide polymorphisms of candidate genes related to coronary heart disease susceptibility by using support vector machines. *Clin Chem Lab Med*, 41:529–534, Apr 2003.

Chapter 3

A Generalized Approach for Testing the Association of a Set of Predictors with an Outcome: A Gene Based Test

3.1 Introduction

In many statistical problems one desires to relate a set of variables to an outcome. For example, it is typical in the social sciences to have data on race, income, education etc. and want to draw inference about the relationship between some outcome and socio-economic status (SES). SES itself is not observed but is instead a combination of the aforementioned variables (as well as others). Some approaches for answering such a question include F-tests and likelihood ratio tests, Fishers's method for combining p-values [Fisher, 1948], and principal components regression. In addition to these generalized approaches, many discipline specific measures have been developed. For example, in psychology it is common to come up with "scores" on different survey instruments. While useful all of these methods suffer from two primary limitations. Firstly, they often rely on parametric modeling assumptions and secondly they often do not take into account the complex relationships of the variables.

Since the underlying relationship between a set of variables and an outcome is usually quite complex and unknown, ideally, instead of specifying a model relating the set of variables to the outcome one would be able to search for the best relationship. Typical statistical learning and prediction methodology is well suited for solving such problems. Statistical learning algorithms apply a basis function (or set of basis functions) to the data to find the best relationship to the outcome. While the best algorithm will depend on the true relationship between the predictor variables and the outcome, most are well suited for situations where the relationship is complex

and/or of high dimension. Typically the focus is on trying to get the best estimate of the function relating the data to the outcome, but in recent years there has been a growing emphasis on variable importance (VI). However, most of the VI measures are ad-hoc and do not have sound statistical properties with a clear parameter of interest, though there is some work trying to formalize VI and attach statistic properties for more targeted analyses [van der Laan, 2006]. Moreover, like with inferential statistics there is not a best means to relate a group of variables to an outcome.

The goal of this paper is to establish a statistical test for assessing the relationship between a set of variables and an outcome. Using tools from statistical learning, a general function is estimated. Others have assessed this relationship using a full permutation test (e.g. Radmacher et al. [2002], Birkner et al. [2005], Chaffee et al. [2010]), however, for all but the simplest algorithms, this is computationally infeasible. Instead, a simple to calculate statistic is proposed. The parameter to be evaluated is the risk between a predicted value and the observed value. This observed risk is compared to an expected risk via a Wald test. A rejection of the null hypothesis that the observed risk is less than the expected risk indicates that the prediction is better than would be expected by chance and that the set of predictors are related to the outcome.

The need for such a statistic can be motivated from two different perspectives: inferential statistics and statistical learning. From an inferential statistics perspective, such an approach can be used to test the association of a group of variables and an outcome (as motivated in the outset). Typically, the group of variables represent an unobserved construct such as SES, a gene, or a stock index. From a statistical learning perspective, the current statistical test represents a means to test whether the prediction derived from a machine learning algorithm is better than what would be expected by chance. This is a question that is not typically asked within the machine learning literature as practitioners generally assume that the set of predictors is related to the outcome and a *significant* prediction can always be derived. While this is not an unreasonable assumption, with the growing ubiquity of prediction methodology, these algorithms are more often being applied to data that may not actually have predictive power (this is particularly true within genetic epidemiology). Therefore, the current method should have broad interest and applicability to both those who are primarily interested in inference as well as those that are primarily interested in prediction.

The paper is organized as follows. In section 2 some preliminaries on statistical learning and loss based estimation via cross-validation are presented. In section 3 the statistical test is proposed with discussion of how to estimate each of the parameters. The next section provides a brief overview of related literature. Section 5 shows some basic simulation results. Section 6 presents an application to genetic epidemiology data. Section 7 provides some concluding thoughts.

3.2 Preliminaries

We begin with the observed data $W_i = (Y_i, X_i) \sim P_W, i = 1, \ldots, n$. The Y_i are the outcome of interest and $Y_i \in \Re^1$. The Y_i can be continuous or binary. The X_i are the covariates, a $p-$dimensional vector, where $X_i \in \Re^p$. We relate X to Y by the functional transformation $f(\cdot)$:

$$\begin{aligned} E(Y|X) &= f(X) \\ P(Y = 1|X) &= f(X) \text{ for } Y \in \{0, 1\} \end{aligned} \tag{3.1}$$

In (3.1) we make no assumptions about the function form of $f(\cdot)$.

3.2.1 Statistical Learning

Statistical learning is concerned with estimating $f(\cdot)$ for the purposes of predicting future outcomes based on an observed covariate vector. All statistical learning algorithms provide a different means of Wrt to Paul's work it is definitely a more computationally friendly version b/c it doesn't require a complete permutation. I think it can also be viewed as an extension of yours and Sandrine's workestimating this function. Table 3.1 lists a range of different learning algorithms. Each algorithm applies a different type of basis function to the data. All algorithms also have a different set of tuning parameters (many have multiple). Changing the tuning parameters optimizes the algorithm for the specific data problem.

Learner	**Type of Function**	**Tuning Parameters**	**Speed**
Regression	Linear Relationship	Variables in Model	Fast
Lasso/Ridge Regression	Penalized Regression	Penalty	Moderate
Nearest Neighbors	Classification based on Proximity	Number Neighbors	Fast
CART	Tree	Tree Depth	Moderate
Splines	Piecewise Functions	Knots	Fast
Support Vector Machines	Transformation of Output Space	Transformation	Slow

Table 3.1: Different Learning Algorithms

Not all algorithms are appropriate for all data problems. For example, if there is a lot of additive structure in the data a linear algorithm will do much better than a tree based algorithm. Conversely, if there are many interactions, then a tree based algorithm would be preferable. Since the choice of best algorithm is dependent on the true underlying function, $f(\cdot)$, which is unknown, it is impossible to know which is best to use.

With this limitation in mind, as computational power has increased, there has been a growing use of ensemble based learners. Ensemble learning is a process of combining multiple learners (typically weak ones) together into one meta learner. There are many different types of ways to ensemble algorithms with the primary methods being: bagging, boosting, Bayesian model averaging and stacking. Ensemble algorithms differ in what their base learners consist of and how the algorithms are combined (i.e. the weights placed on each algorithm). For example the Random Forests algorithm [Breiman, 2001] is an ensemble based algorithm where the base learners are unpruned CART trees and they are combined via bagging, a process of adding equal weight to all learners.

Different ensemble algorithms will have different strengths and weaknesses. In the present work, the goal is to use the algorithm that best estimates the underlying function $f(\cdot)$. The algorithm that has been found to be most adapted for this problem is the SuperLearner (SL) algorithm [van der Laan et al., 2007]. SL is an algorithm based on stacking [Wolpert, 1992]. In stacking based algorithms a library of algorithms is applied to the data and each algorithm provides a predictions of the outcome via cross-validation (CV). The predictions for each observation are *stacked*, creating a matrix of predicted outcomes for each of the j learners. The true outcome is regressed onto the predicted values. The derived coefficients provide the weights for each algorithm.

In the implementation of SL, available in SuperLearner package in R, the regression performed is a non-negative least squares and the coefficients are scaled to sum to 1. Other authors [Breiman, 1996, Ting and Witten, 1997] have similarly found non-negative least squares to be the optimal majorizing function. While typical implementations of stacking involve using similar base learners with different tuning parameter settings, van der Laan et al. advocate using a full library of different types of learners covering a range of basis functions. The authors were able to show that stacking satisfies certain oracle properties which we repeat here:

Oracle Inequalities: Let $d_o(\psi, \psi_0) = E_{P_X}\{L(X, \psi) - L(X, \psi_0)\}$ be the risk difference between the candidate estimate ψ and the true parameter value ψ_0. Also, suppose the $P\{(\hat{\Psi}_k(P_n) \in \Psi) : \forall k\} = 1$. Assume:

A1: $L(X, \psi)$ is uniformly bounded

A2: The variance of the ψ_0-centered loss function $(L(X, \psi) - L(X, \psi_0))$ can be uniformly bounded by its expectation uniformly in ψ.

then, for any $\lambda > 0$:

$$Ed_0(\Psi_{\hat{K}(P_n)}(P_{n,T(V)}), \psi_0) \leq (1 + 2\lambda)Ed_0(\Psi_{\tilde{K}(P_n)}(P_{n,T(V)}), \psi_0) + 2C(\lambda)\frac{1 + \log(K(n))}{np}$$

where p is the proportion of the observations in the validation sample and $C(\lambda)$ is a constant defined in van der Laan et al. [2006].

These results imply that the SL performs as well as the oracle selector in terms of expected risk difference and as long as the number of candidate learners ($K(n)$) is polynomial in sample size, the SL is the optimal learner. Moreover, if one of the candidate learner searches within a parametric model and that model contains the truth, then the SL attains an almost parametric rate of convergence $\log n/n$. This makes the SL an ideal learner when the true underlying function is unknown.

In practice, implementing the SL is fairly straightforward. The main tuning parameters include selecting the candidate library, the number of CV splits, and the majorizing function. The majorizing function, as mentioned, is typically non-negative least squares. While the larger the number of CV splits the better the estimate of the function, there is obviously a computational trade off. Typically 10-fold or 20 fold CV has been found to be appropriate. The most important aspect is the candidate library. Again, while one may say "the more the better" (up to a limit), in practice fitting each candidate can be highly computational and it is worth being judicious in the choice of candidate learners. Generally, then, it is best to apply SL with a library that spans a range of basis functions. One final note about the computation is that it is fairly straightforward to implement the SL either within a cloud environment or across nodes in a parallel environment.

3.2.2 Loss Based Estimation via Cross-Validation

Once the learning algorithm is fit to the data a prediction, $\hat{f}(X_i)$, is generated for each observation, Y_i. The goal in prediction is to minimize the risk over the training set:

$$argmin_f E[L(Y, \hat{f}(X)] \tag{3.2}$$

The "harder" $\hat{f}(\cdot)$ is fit to the data, the greater the potential for over-fitting, referred to as the *optimism*, and, consequentially under-estimating the risk [Hastie et al., 2009]. There are two main approaches for correcting for over-fitting. The first is by directly estimating the optimism and adding this to the estimated training error (e.g. AIC, BIC, MDL). The other is to directly estimate the test error (e.g. CV, Bootstrap methods). Direct methods are useful when the number of basis functions (effective number of parameters) are easily calculable (e.g. linear models, regularized regression). For more complex methods (including all ensembles), such calculations are intractable. CV methods provide the simplest approach to obtaining an honest estimate of $\hat{f}(X)$, and consequently the risk.

In all CV methods, the data are divided into a *training* and *validation* set. The estimator is computed (*trained*) on the training set and then tested (*validated*) on the remaining validation set. This process is iterated, allowing each observation to be part of the validation set, providing an unbiased estimate of the risk of the estimator [Dudoit and van der Laan, 2005].

Using notation from Dudoit and van der Laan, we define a binary random vector, $B_n = (B_n(i) : i = 1, \ldots, n) \in \{0, 1\}^n$, independent of the empirical distribution P_n. A realization, $B_n(i)$ represents an indicator of whether an observation is in the training or validation set:

$$B_n(i) = \begin{cases} 0, & i\text{th observation } X_i \text{ is in the } \textit{training} \text{ set}, \\ 1, & i\text{th observation } X_i \text{ is in the } \textit{validation} \text{ set}, \end{cases} \tag{3.3}$$

Let P^0_{n,B_n} and P^1_{n,B_n} denote the empirical distributions of the training and validation sets, respectively, and let the number and proportion of observations in the validation sets be denoted by $n_1 \equiv \sum_i B_n(i)$ and $p = p_n \equiv n_1/n$, respectively. Then a definition of the cross-validated risk estimator for $\psi_n = \hat{\Psi}(P_n)$ is

$$\begin{aligned} \hat{\theta}_{p_n,n} &\equiv E_{B_n} \Theta(\hat{\Psi}(P^0_{n,B_n}), P^1_{n,B_n}) \\ &= E_{B_n} \int L(x, \hat{\Psi}(P^0_{n,B_n})) dP^1_{n,B_n}(x) \\ &= E_{B_n} \frac{1}{n_1} \sum_{i:B_n(i)=1}^{n_1} L(X_i, \hat{\Psi}(P^0_{n,B_n})) \end{aligned} \tag{3.4}$$

where $\hat{\Psi}(P^0_{n,B_n})$ represents the estimator of the parameter ψ based on the training set.

The distribution of B_n determines the type of CV. The most common type of CV is *V-fold cross-validation*. In V-fold CV the learning set is randomly divided into V mutually exclusive sets of approximately equal size. Each set is then used in turn as a validation set (see figure 3.1). The distribution of B_n places mass $1/V$ on each of V binary vectors such that $\sum_i b^v_n(i) \approx n/V \; \forall \; v$ and $\sum_v b^v_n(i) = 1 \; \forall \; i$. Other forms of cross-validation include *Leave-one-out cross-validation*, *Monte Carlo cross-validation* and *Bootstrap-based cross-validation* [Dudoit and van der Laan, 2005].

Dudoit and van der Laan showed that the risk estimate provided through CV is asymptotically linear with appropriate assumption (see Theorem 3 in their paper) and has influence curve (IC):

$$\begin{aligned} IC &\equiv L[Y, f(X)] - \theta \Rightarrow \\ \hat{\theta} &\cong \frac{1}{n} \sum_{i=1}^{n} L[Y_i, \hat{f}(X_i)] - \hat{\theta} \end{aligned} \tag{3.5}$$

The benefit of defining the IC of a parameter is that one can use the variance of the IC to obtain the variance of the estimator. Dudoit and van der Laan use this result

Figure 3.1: Illustration of 4-fold CV where $n = 8$ and $n_1 = 2$. In each cycle, 6 observations are used to train the learner and 2 are used to test or validate it. Courtesy of maxdama.com

to construct confidence intervals for the risk estimate

$$\hat{\theta}_n \pm z_{1-\alpha/2}\frac{\sigma_n}{\sqrt{n}}$$

where $\hat{\theta}_n$ is the estimated cross-validated risk and σ_n is the standard deviation of the cross-validated loss. Asymptotically the observed standard deviation, $\hat{\sigma}$, is an appropriate estimator. Bengio and Grandvalet [2004] showed that while there is no unbiased estimate for σ in finite samples, it does converge to the observed standard deviation fairly rapidly (see Section 3.4 for discussion).

3.3 The proposed test statistic

The parameter of interest, θ, is the risk from our prediction algorithm:

$$\begin{aligned} \theta &\equiv EL(Y, f(X)) \\ \hat{\theta} &\equiv EL(Y, \hat{f}(X)) \end{aligned} \tag{3.6}$$

based on the model defined in (3.1). While Dudoit and van der Laan [2005] focused on generating confidence intervals for the observed risk, $\hat{\theta}$, the present interest is in hypothesis testing. We can test the hypothesis:

$$\begin{aligned} H_o &: \theta \geq \theta^* \\ H_a &: \theta < \theta^* \end{aligned} \tag{3.7}$$

This is a one-sided hypothesis test of whether the observed risk, $\hat{\theta}_n$ is less than some expected risk under a null hypothesis, θ_n^*. We define

$$\theta^* \equiv E[L(Y, f(X))] s.t. Y \perp X \tag{3.8}$$

We can then use the same asymptotic linearity result and define a Wald-type statistic, with parameter ψ:

$$\begin{aligned} Z &= \frac{\hat{\psi}\sqrt{n}}{\sqrt{v\hat{a}r(IC(W;\psi)}} \\ &= \frac{\hat{\theta}_n - \hat{\theta}_n^*}{\sqrt{var(\hat{\theta}_n - \hat{\theta}_n^*)}} \sim N(0,1) \end{aligned} \tag{3.9}$$

This will be a one-tailed test as we are only interested in the case where $\hat{\theta}_n < \hat{\theta}_n^*$.

From (3.9) there are three values that need to be estimated:

- The estimated risk: $\hat{\theta}_n$
- The estimated risk under the null: $\hat{\theta}_n^*$
- The variance of the difference of the two: $var(\hat{\theta}_n - \hat{\theta}_n^*)$

3.3.1 The Observed Risk

The observed risk is the simplest value to estimate. Whereas any loss function can be used, a loss that has certain asymptotic properties will be needed to allow for the use of the IC to calculate the asymptotic variance. Based on the work of Dudoit

and van der Laan [2005], this includes most any loss, with the notable exception of misclassification loss.

For mathematical simplicity, that will be seen later, squared error (ℓ_2) loss is used, with the estimated risk being:

$$\hat{\theta}_n \equiv E_n L(Y, \hat{f}(X)) \equiv E_n (Y - \hat{f}(X))^2 \tag{3.10}$$

3.3.2 The Null Risk

The null risk is the expected value of the loss between the observed outcome and the predicted outcome, when the set of covariates, X, is independent of the outcome, Y. The most direct way to estimate this is by permuting the X values and retraining the predictor. For all but the simplest prediction algorithms, though, this can be computationally infeasible. However it is possible to estimate this value. We need:

$$\theta(P^*) = E_{P^*}[Y - f(X)]^2$$

Assuming that f(X) is of fixed form, and the learning algorithm has an intercept:

$$Ef(X) = \mu_Y = EY$$

Therefore, we get:

$$\begin{aligned} &E_{P^*}[(Y - \mu_Y + \mu_Y - f(X))^2] \\ &= E_{P^*}(Y - \mu_Y)^2 + E_{P^*}(f(x) - \mu_Y)^2 + 2E_{P^*}\big((Y - \mu_Y)(\mu_Y - f(X)\big) \end{aligned}$$

So:

$$\begin{aligned} \theta &= var(Y) + var\big(f(X)\big) - 2cov\big(f(X), Y\big) \\ \theta^* &= var(Y) + var\big(f(X)\big) \end{aligned} \tag{3.11}$$

Thus a test of the null that $\theta = \theta^*$ is a test of the $cov\big(f(X), Y\big) = 0$.

3.3.3 The Variance of the Difference of Risks

The final consideration is defining the variance of the difference of $\hat{\theta}_n$ & $\hat{\theta}_n^*$. Using the property of ICs we have:

$$var(\hat{\theta} - \hat{\theta}^*) = \frac{var(IC[\hat{\theta}] - IC[\hat{\theta}^*])}{n} \tag{3.12}$$

Therefore we need to define the IC for $\hat{\theta}$ and $\hat{\theta}^*$. We note the parameter of interest can be expressed as:

$$\psi \equiv E[Y - f(X)]^2 - E[Y - EY]^2 - E[f(X) - E(f(X))]^2 \tag{3.13}$$

This is simply the difference of three loss functions. Dudoit and van der Laan [2005], provided the framework for calculating the influence curve for any general loss function, noted in (3.5). Therefore:

$$\begin{aligned} IC(W;\psi) &= IC_1 - IC_2 - IC_3 \\ &= \big(L[Y, f(X)] - \theta\big) - \big(L[Y, EY] - \theta_Y\big) - \big(L[f(X), Ef(X)] - \theta_{f(X)}\big) \end{aligned} \tag{3.14}$$

By substituting the appropriate loss function (in this case ℓ_2) and taking the variance of (3.14) one can calculate the inference for the test statistic. There are now all of the elements of the test statistic and referring back to (3.9) we can write:

$$\begin{aligned} Z &= \frac{\hat{\psi}\sqrt{n}}{\sqrt{v\hat{a}r(IC(W;\psi)}} \\ &= \frac{\hat{\theta}_n - \hat{\theta}^*_n}{\sqrt{var(\hat{\theta}_n - \hat{\theta}^*_n)}} \\ &= \frac{\sqrt{n}\Big[E_n(Y-\hat{f}(X))^2 - E_n(Y-\bar{Y})^2 - var(\hat{f}(X))\Big]}{\sqrt{var\Big(E_n[(Y_i-\hat{f}(X_i))^2+\hat{\theta}] - E_n[(Y_i-\mu)^2+\hat{\sigma}^2_Y] - E_n[(\hat{f}(X_i)-E\hat{f}(X_i))^2+\hat{\sigma}^2_{\hat{f}(X)}]\Big)}} \end{aligned} \tag{3.15}$$

The statistical test as constructed will be unbiased for predicting on an independent sample. However, while asymptotically the proposed test statistic will approach $N(0,1)$ in practice the estimated variance a bit too small. This is due to the excess correlation induced by cross-validation (see [Bengio and Grandvalet, 2004]). Simulations showed that the asymptotics do not fully kick in until extremely large samples ($n > 1 \times 10^7$). Grandvalet and Bengio [2006] proposed a finite sample correction (see section 3.4). Based on experimental results it was found that the variance in (3.15) underestimated the true variance by a factor of 2 (see figure 3.2 in section 3.5). This is obviously a less than ideal solution and current work involves trying to determine a more theoretical finite sample correction.

To calculate the statistic in R code:

```
sqLOSS <- function(x,y)(x-y)^2

predTestL2 <- function(pred,Y,LOSS = sqLOSS){
  n <-length(Y)
  ll <- LOSS(pred,Y)  ###\hat\theta
```

```
  lo <- LOSS(mean(Y),Y) - LOSS(pred,mean(pred)) ###\hat\theta^*
  LossD <- mean(ll - lo)
  varIC <- var(ll - lo)*2
  Z <- LossD*sqrt(n)/sqrt(varIC)
  return(Z)
}
```

3.3.4 A Permutation Based Test

It is also possible to construct a significance test via the permutation distribution. To test the independence of Y and the p-vector X one could permute the Y and continually retrain the predictor [Radmacher et al., 2002]. However, for the most part this is too computational. Alternatively, one can test the independence of Y and $\hat{f}(X)$. To do so:

1. Train the predictor to obtain $\hat{f}(X)$
2. Calculate $\hat{\theta} = \sum_i L(Y_i, \hat{f}(X_i))$
3. Permute the Y_i b times and calculate $\hat{\theta}_j^* = \sum_i L(Y_i^*, \hat{f}(X_i))$ for $j \in 1 \dots b$
4. The permutation based p-value is $\frac{1}{b} \sum_j I\left[\hat{\theta} < \hat{\theta}_j^*\right]$

The choice of b will depend on the desired precision of the empirical p-value, with stronger associations requiring larger b. To calculate this in R code:

```
predTestPerm <- function(pred,Y, p = 1000, LOSS = sqLOSS){
  Yp <- replicate(p, sample(Y))
  mZ <- median(LOSS(pred,Y))
  Zp <- apply(Yp,2,function(x)median(LOSS(pred,x)))
  pval <- 1 - sum(mZ < Zp)/p
  return(pval)
}
```

3.4 Previous Work with Assessing Prediction

In a general sense, the proposed test can be considered as analogous to an F-test, used in linear regression. Both approaches aim to test the goodness-of-fit of a fitted function based on the residual fit. The primary distinction is that while an F-test relies on the correctly specifying the parametric form of $f(\cdot)$, the proposed test can be seen as a semi-parametric alternative that does not depend on a specified functional form (see Figure 3.5 in Section 3.5).

There has been some related work assessing the significance of a prediction. The work of Dudoit and van der Laan [2005] focused on providing a theoretical basis for calculating the standard-error of the CV risk estimate for the purpose of constructing confidence intervals. Their work focused on the asymptotic properties, showing that it is both consistent and asymptotically linear.

Bengio and Grandvalet [2004] were also interested in constructing confidence intervals for the CV risk estimate, however they focused on finite samples. The authors showed both theoretically and via simulation that in finite samples it is not possible to get an unbiased estimate of the variance of the cross-validated risk estimate. They broke down the variance into three components:

(1) The variability of the prediction within each validation block

(2) The covariance between predictions within each block

(3) The covariance between predictions in different blocks

The first value is the quantity of interest, however, simply taking the empirical variance of $\hat{f}(X)$ is biased by the other two quantities. However, the authors showed, with modest sample sizes ($n > 100 - 500$) these two values go to 0, resulting in the desired value. The authors proposed a corrected test statistic based on an assumed maximum between block correlation of 0.7 [Grandvalet and Bengio, 2006]. This correction is similar to the proposed correction in the current work. The primary limitation of their formulation is that they construct a t-test against a fixed value. However, as shown above, if one wants to test against an expected risk, it is necessary to estimate θ^* and therefore the variance of the estimate also needs to be considered.

Dietterich [1998] showed also that it is only in small samples one needs to be concerned with the variance estimates of the cross-validated risk. This work provides further finite sample justification for the work of Dudoit and van der Laan and much of the present discussion.

Other work includes Radmacher et al. [2002] who laid a general framework for assessing the prediction in micro-array studies. The authors advocated permuting and then refitting the learner, to assess the risk estimate via cross-validation. However, they were working with much simpler learners. Others have used this full permutation approach for testing genetic pathways [Birkner et al., 2005] and perfoming gene set tests [Chaffee et al., 2010]. Lusa et al. [2007] noted that simply calculating the odds-ratio on a two-by-two table led to inflated type I error. This was also noted by Lee [2007]. This observation conforms to the theory presented in Dudoit and van der Laan, which showed that misclassification loss is not an appropriate loss function to use for the present work.

3.5 Simulations

To examine the behavior of the test statistic a series of simulations were undertaken. It is re-noted that the Z-test is a one-tailed test where the more negative the test statistic, the more significant the association. The first simulation was aimed at examining the need for a correction of the test statistic. Figure 3.2 shows the true variance of $\hat{\theta} - \hat{\theta}^*$ compared to both the asymptotically calculated variance as well as the corrected variance. One-thousand simulations were performed using full terms regression as the only learner, and 10-fold CV (larger folds of CV were performed and did not impact the results). Both the sample size and the number of parameters in the model were varied. Results suggest, that the ratio between the true variance and the asymptotic variance is fairly consistent at 2. The absolute variance decreases as sample size increases and increases with the number of parameters. The great decrease in the absolute difference of the expected and estimated variance with large sample, suggests that the asymptotics are working, but the relative difference, indicates that the correction is of value.

Once determining an appropriate correction for the test statistic, the next series of simulations aimed to understand the statistic itself. The first simulation explored the null situation where $Y \sim N(0,1)$ was independent of $X \sim N(0,1) \in \Re^p$. One-thousand simulations were run, using 10 predictors, under three different sample sizes $(100,\ 2,000,\ 10,000)$. Linear regression was the only model used to estimate $f(X)$. As the sample size increases the test statistic becomes more normally distributed. Moreover the correction becomes less important. For the permutation test, a Z-quantile for the empirical p-value was calculated (see Figure 3.3).

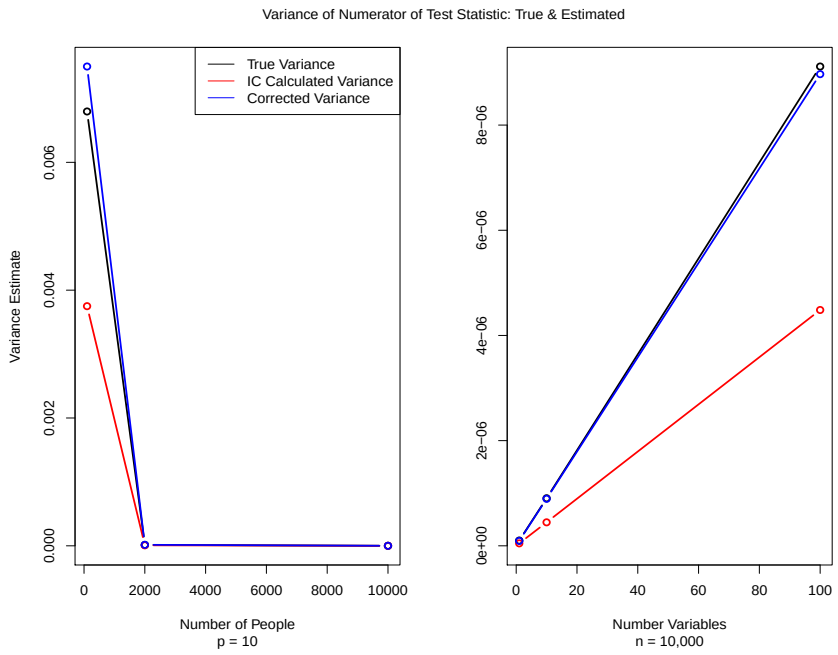


Figure 3.2: The true variance and estimated variance of $\hat{\theta}_n - \hat{\theta}_n^*$ across 1,000 simulations. In the left hand figure, the number of parameters is fixed at 10 and the number of people is varied from 100 to 10,000. In the right hand figure the number of people is fixed at 10,000 and the number of parameters is varied. Unsurprisingly the variance decreases with sample size and increases with the number of parameters. Of greater interest, the ratio between the asymptotically estimated variance and the true variance remains relatively constant at 2, though decreases in absolute terms with increasing sample size. Finally, the increase in variance due to the number of parameters appears to be linear in p.

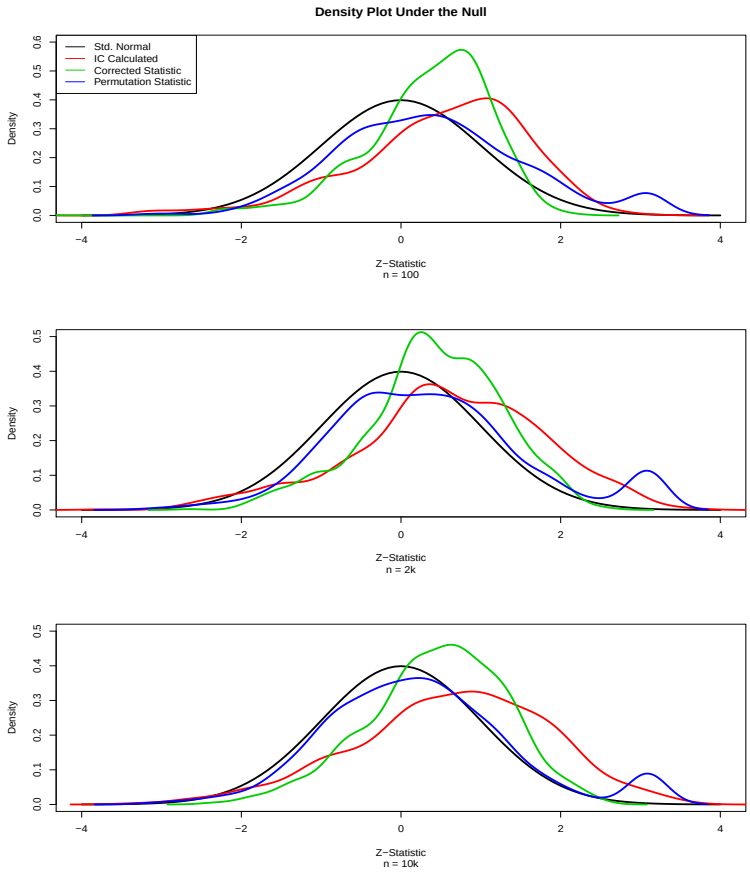


Figure 3.3: Behavior of test statistic with increasing sample size. One-thousand simulations performed with continuous X and Y and no association between them. 10 predictors are used. For larger n both the asymptotic and the corrected test statistics become more normal with variance 1. The uncorrected is slightly anti-conservative in the extreme (negative) tails highlighting the need for the correction at lower sample sizes. The permutation p-value have been transformed to a normal distribution for comparison and maintains appropriate error control.

Secondly, the test statistic was examined under an *alternative* scenario where there is a true relationship between Y & X. In this simulation, p = 10, and different sample sizes were used. $Y = X_1 + \epsilon$, with $Y \perp X_2 \dots X_{10}$. For simplicity, the corrected test statistic was calculated. Figure 3.4 shows the simulation results. While, the test has little power in low sample size (n = 100), the test gains power as the sample size increases, and maintains a $N(\mu, 1)$ distribution.

To illustrate the comparison to the F-test, two more simulations were undertaken, both under an alternative model. In the first, $Y = X_1 + \epsilon$ again. In the second, $Y = X_1 * X_2 + \epsilon$, an interaction between two of the X covariates but with no main effects. To calculate the F-statistic the full main effects model was fit. In scenario I, it is expected that the F-statistic should capture the Y, X_1 relationship. However, in simulation II, the model is now misspecified. To calculate the proposed test, a SuperLearner was fit, using a step wise algorithm and an intercept function.

The average p-values for each scenario are calculated and presented in figure 3.5. Both methods are able to detect the association when there is a main effect term in the model. However, once there is only an interaction, the F-test loses all of its power due to the misspecified model. The SuperLearner approach is able to flexibly search for the best model and consequently has ample power to detect the association. It is thus flexible against semi-parametric alternatives, giving this approach its power.

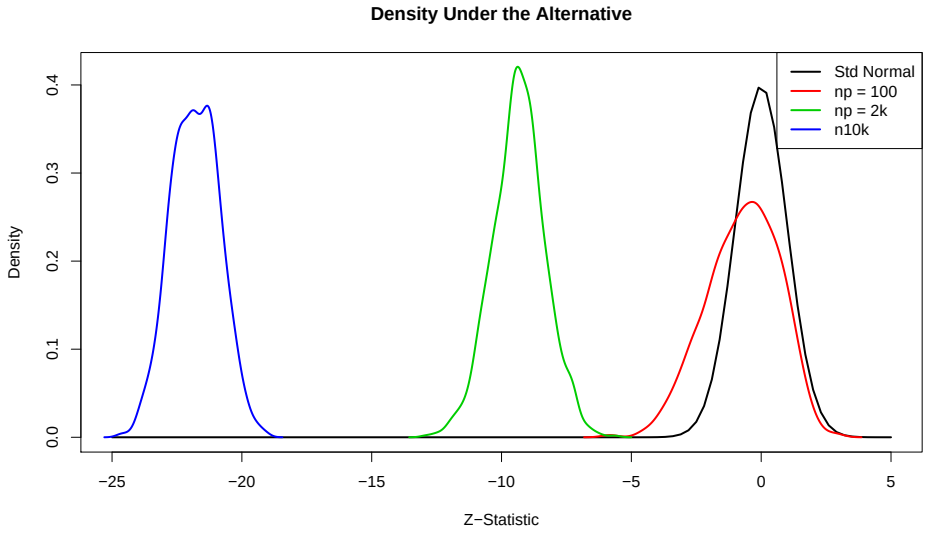


Figure 3.4: Distribution of the test-statistics when there is an association. Again linear regression is the only model used to estimate $f(X)$. The corrected test-statistic is calculated. As the sample size increases, the test statistic becomes both more significant (more negative) and approaches $N(\mu, 1)$.

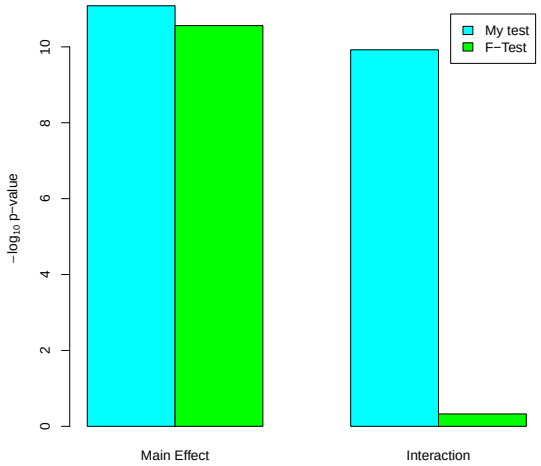


Figure 3.5: Average $-\log_{10}$ of the p-values for the two "alternative" simulation settings. In the first, there is a a true main effect term in the model and both the F-test and the proposed test are able to detect it. In the second, there is only an interaction, and the miss-specified regression model is not able to detect the association, while the more flexible SuperLearner based method is able to find the right model and detect the association.

3.6 Application to genetic data

A typical means of studying the genetic causes to diseases is via SNP association studies. This design involves recruiting 1000's of people with and without a disease of interest, referred to as cases and controls respectively. Each individual is typed on a set of single nucleotide polymorphisms (SNPs). Each SNP represents a single base pair of DNA where there is a degree of variation across the population (most DNA is fixed and does not vary). Genes are made up of 100s or 1000s of base pairs of DNA. Amongst these bases there will be dozens or 100s of SNPs. In Genome Wide Association (GWA) studies individuals may be typed on up to 1 million SNPs across the genome, aiming to capture ones common genetic variation (on the nucleotide level). In more focused candidate gene studies individuals may be typed on 10's of thousands of SNPs aimed at well characterizing specific genes. These studies represent an a hypothesis-free search across the genome for regions of interest to be followed up on.

For the most part GWAs have been successful at identifying SNPs, and be extension genes, associated with many common diseases [WTCCC, 2007]. However, it is not presumed that any associated SNP is itself causal. An associated SNP may be correlated (referred to as in linkage disequilibrium [LD]) with the true causal variant, located within or near the same gene. Therefore all results need to be confirmed via replication. Moreover, since the studies are initially exploratory (until replication has occurred) the true unit of interest the gene in which the SNP lies. This has led towards the recognition of the need for gene based tests [Neale and Sham, 2004].

In recent years there has been growth of gene based tests of association (see Beyene et al. [2009] for a recent review). These methods can roughly be broken down into (i) clustering and PCA based approaches and (ii) logistic modeling and combining marginal p-values. The method used in the popular and freely available software PLINK [Purcell et al., 2007], is a variation of Fisher's Method [Fisher, 1948] that uses the permutation distribution to assess significance. However, most of these methods suffer from one primary limitation: they rely on marginal p-value (as calculated by a χ^2 test) to assess the gene based association. While marginal testing has been somewhat successful in detecting associations, and is (more importantly) computationally simple, it does not well capture complex associations. SNPs may interact or be involved in complex joint associations with other SNPs [Heidema et al., 2006]. Therefore, methods dependent on typical marginal tests may be ill suited for creating gene based tests.

One can consider the proposed to test as another means of performing a gene based test for association. In this setting, the observed units would be the SNPs and the unit of interest is the gene which they comprise. A prediction model relating the SNPs in a gene to disease status serves as a test for association for the entire gene. This point is illustrated first via simulation and then in application to a candidate gene study.

3.6.1 Simulation Study

A more comprehensive simulation study was undertaken to explore the use of the test statistic for candidate gene studies. Fifty simulations were performed. In each simulation 440 *genes* were simulated, consisting of 10 *SNPs*. Each of the 10 *SNPs* were independent and had a minor allele frequency of 0.3. In this sense a realistic genetic structure was not simulated, where one would expect complex correlation between SNPs and varying minor allele frequencies. However, to explore the performance of the method this was not necessary.

Of the 440 genes in each dataset, 40 (10%) of the genes were associated with the outcome. The goal of the simulation was specifically explore how the proposed test compares to standard methods when there is a complex association. Four different association models were used:

Additive: $P(D|SNPs_{Gene}) = \beta(X_1 + X_2 + X_3)$ with $X_i \in \{0, 1, 2\}$

Dominant: $P(D|SNPs_{Gene}) = \beta(X_1 + X_2 + X_3)$ with $X_i \in \{0, 1/2\}$

Recessive: $P(D|SNPs_{Gene}) = \beta(X_1 + X_2 + X_3)$ with $X_i \in \{0/1, 2\}$

Interaction: $P(D|SNPs_{Gene}) = \beta(X_1 * X_2)$ with $X_i \in \{0, 1, 2\}$

These represent genic style associations where multiple *SNPs* within a *gene* lead to an increase in the probability of disease.

For each dataset three measures of association were calculated. First, the marginal association for each of the $4,400$ *SNPs* was calculated via the allelic χ^2-test. This is the typical test for association in genetic epidemiology studies. It is a 1-df test that compares the frequency of the alleles between those with disease and those without. The second measure of association was the variation of Fisher's Method for combining p-values. To calculate the p-value, 10,000 permutation were performed. Finally, the current test was used to estimate the function

$$P(D|SNPs_{Gene}) = f(SNPs)$$

A SuperLearner was fit using a library of: k-Nearest Neighbors, a logistic regression step function, RandomForests, LASSO and an intercept function, using 10 fold CV to both fit and validate the function. The corrected parametric statistic was used to calculate the p-value.

For each method the false discovery rate (FDR) was controlled using the Benjamini-Hochberg (BH) [Benjamini and Hochberg, 1995] procedure at a level of 5%. The average power and error-rate across the 50 simulations was calculated for each procedure. For the marginal testing, if one of the *SNPs* in the *gene* passed the significance threshold, then the entire gene was declared significant.

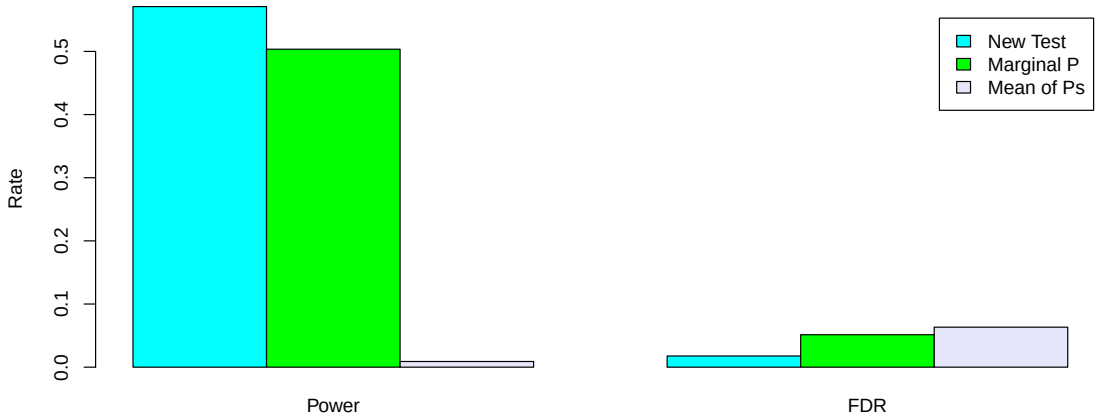


Figure 3.6: Bar plot comparing three methods for assessing the association of a *gene*. The proposed test has both the greatest power ($\sim$ 57%) lowest FDR ($\sim$ 1.8%) The FDR was controlled at a level of 5%. Marginal testing proved to be fairly successful, though the greater multiple testing burden incurred by the 10-fold increase in the number of tests makes it less powerful. Fisher's Method was not able to maintain a high significance level.

Figure 3.6 shows the power and FDR for the three methods. The proposed test has the most amount of power (57% vs. 50% & 1%) as well as the lowest error-rate. While marginal testing was fairly successful, the extra multiple testing burden incurred by the extra tests decreased its power. It should be noted, that in an actual association study, this burden would be even greater as most genes have many more than just 10 SNPs. Finally, Fisher's Method was least successful, this owes to the duel fact that it only looks at marginal associations and considers all tests simultaneously, while the proposed method, looks at the joint effects of only those tests of importance.

These results should not necessarily be interpreted that the proposed test is uniformly more powerful than these or other methods. The proposed statistic is an omnibus test, and only a few types of causal models were explored. For example, if only one *SNP* for a given *gene* were causal, then the marginal test would be most powerful. Likewise, if most of the *SNPs* for a given *gene* were associated than Fisher's Method would have more power. The value of the present statistic, though, is that it does not require one to specify a specific model, but search for the best fitting one, while still maintaining adequate power.

To illustrate this a second simulation was undertaken. As opposed to a complex

association, only one *SNP* was associated per *gene*. Moreover, the marginal p-value was simulated to be approximately 1×10^{-8}, the standard cut-off for genome wide significance. One thousand *genes* were simulated, and the median the p-value for the associated *SNP* was 8.5×10^{-9}. The median p-value on the gene-based test, was 1.6×10^{-4}. This shows that even when the causal mechanism favors the marginal test, there is still ample signal to find an association via this more flexible approach. For more complex associations, as well as for larger datasets with greater multiple testing burden, this difference will decrease.

3.6.2 Data Analysis

To examine how this methods works with real data, three analyses were performed using a data set derived from a candidate gene study. The first was a typical analysis to determine which genes were associated with disease. The second was aimed at exploring the association of a genetic pathway. The third is a more unique cluster based analysis.

Data for all analyses comes from a 2007 candidate gene study from the International Multiple Sclerosis Genetics Consortium. Multiple Sclerosis (MS) is an autoimmune disease known to have a strong heritable component based on epidemiological studies [Oksenberg and Barcellos, 2005]. The major histocompatibility complex (MHC) region of chromosome 6 has long been known to be associated with MS, however few other genes have been definitively identified.

The goal of the 2007 was to follow-up on suspected MS genes. The data collection has been described previously [IMSGC, 2010]. In brief, the data consisted of $1,379$ controls and $1,343$ cases. Data were collected on $52,801$ SNPs across 9552 genes. After data cleaning there were $46,057$ SNPs.

Candidate Gene Study

The first analysis was aimed at detecting which genes are associated with MS. All genes that had at least 8 SNPs in them and were not on chromosome X or within the MHC were selected. Chromosome X genes were dropped to avoid gender effects, while MHC genes were dropped since the goal was to detect genes that were not known to be associated with MS. This left $1,254$ genes comprising $25,362$ SNPs.

The three methods for association were calculated as in the simulation study: the generalized method, the mean of the p-values (Fisher's Method) and simple marginal testing. For the generalized method, a SuperLearner was fit to estimate the function using the same candidate learners as above, including also, Support Vector Machines and PolyClass. For the mean of p-values, $10,000$ permutations were used, performed in PLINK [Purcell et al., 2007]. For marginal testing, the minimum p-value for each gene was recorded.

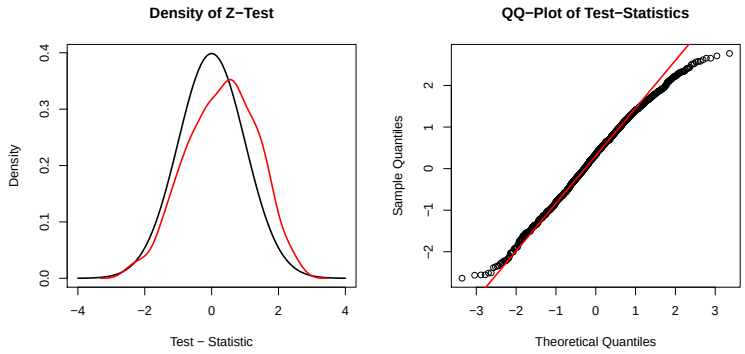


Figure 3.7: Distribution of test statistic for all genes. Left is a density plot while the right is a qq-plot. The plots suggest that there are no greater associations that would be expected by chance.

After controlling for multiple testing using the BH-FDR, none of the measures of association provided an FDR below 10%, suggesting that the smallest observed p-values would be expected simply by chance. Figure 3.7 shows the distribution of the test statistic across the 1254 genes, both as a density plot and qq-plot. This further reflects that even though small p-values were observed, these conform with what would be expected.

While it is not possible to reject the global null of no genes being associated, it is still of value to explore the most significant genes. Table 3.2 lists all the genes that had a p-value less than .01 on the proposed test. Also listed are their p-values for Fisher's Method as well as the smallest marginal p-value. From the table, it is clear that some genes had strong association regardless of the method used. These include STAT4, IL7 and CLEC16A. Unsurprisingly, IL7 and CLEC16A have previously been identified as MS genes [Hafler et al., 2007]. In this sense, the proposed method is able to replicate previous findings. However, the other genes, while showing some marginal association, would not be detected by typical methods. This suggests, that if truly associated, the mechanism is more likely more complex than a single SNP association. Also noteworthy, is that the size of the gene appears to be independent of the strength of association. Both small and large genes had strong associations.

Gene	Chr	Num SNPs	Gene P-val	Mean P-val	Min SNP P-val
STAT4	2	139	0.099	0.006	2.86×10^{-5}
EFNA5	5	10	0.099	0.386	0.056
MAP1B	5	13	0.005	0.379	0.055
IL7	8	73	0.009	4.0×10^{-4}	2.04×10^{-5}
RBM17	10	77	0.009	0.310	0.021
IRF7	11	8	0.006	0.044	3.49×10^{-4}
CLEC16A	16	14	0.005	1.0×10^{-4}	1.08×10^{-5}
MYO1D	17	9	0.004	0.165	0.075
APP	21	9	0.005	0.104	0.044

Table 3.2: All Genes that had a p-value less than .01 on the gene based test. The p-value based on Fisher's Method as well as the minimum marginal p-value is also shown. The genes highlighted in red probably would not have been detected by alternative methods. Other genes, such as IL7 & CLEC16A not surprisingly have previously been associated with MS.

Pathway Analysis

Using the same dataset, a second analysis was undertaken. Instead of looking at individual genes, whole genetic pathways were explored. Many genes have shared biological functions, referred to as genetic pathways. One such pathway of interest for MS is the DNA Repair pathway. The DNA repair pathway consists of four sub-pathways. Briggs et al. [2010] studied the pathways' relationship with MS, using a mixture of machine learning (Random Forests) and parametric modeling (Logistic Regression). The results suggested that the only important gene in the pathway is GTF2H4, which is located within MHC region of chromosome 6. However, the results were not definitive, and using the same data, the same pathways were reanalyzed using the current method.

In pathway analysis the observed unit is still individual SNPs, but the unit of interest is now a collection of genes. Methodologically, the approach is the same as analyzing a gene, except the number of SNPs are increased. A SuperLearner was fit using the same candidates as above. The four pathways were analyzed separately and the results are shown in Table 3.3. The only associated pathway is the NER pathway, which contains GTF2H4. Analyzing GTF2H4 independently, revealed an association very close to the full NER association. Finally, analyzing the NER pathway without GTF2H4 revealed no association. These results are a more definitive confirmation of the findings in Briggs et al. [2010] that GTF2H4 is the only important component of the DNA Repair pathways as it relates to MS.

Pathway	Num Genes (SNPs)	Z-Statistic	P-value
BER	22 (127)	0.12	0.55
HR	15 (124)	-0.29	0.38
NHEJ	9 (90)	-0.97	0.17
NER	26 (208)	-3.12	9.03×10^{-4}
NER (w/out GTF2H4)	197	-0.68	0.25
GTF2H4 only	11	-3.79	7.53×10^{-5}

Table 3.3: Results for the DNA Repair pathway analysis. Only the NER pathway shows any association with MS. However, upon further examination that association is based entirely on GTF2H4 confirming the results in Briggs et al. [2010].

Clustering the MHC

To illustrate the flexibility of this method to genetic data a very different analysis was performed. Instead of testing the association of a region, the predictions were used to cluster genes. As mentioned, the MHC region of chromosome 6 is well known to be highly associated with MS (as well as many other auto-immune diseases). While the gene based analysis revealed that almost every MHC gene had an association (often very strong ones), one question of interest is whether these associations are due to the strong and complex correlation (LD) in the MHC or are independent signals. While HLA-DRA is known to be associated with MS, recent studies have suggested that other genes may also be independently associated (e.g. Cree et al. [2010]).

In order to explore this question a novel approach was taken. The predictions ($\hat{f}(X)$) were calculated for each of the 78 MHC genes in the dataset. Then using the predictions of just those with disease ($n = 1343$) all of the genes were clustered using the partitioning around medoids (PAM) algorithm. PAM is similar to k-means clustering, with the primary difference being, instead of minimizing an average distance, a median distance is minimized, making the findings more robust. The MHC can be divided into three classes. Therefore the number of centers, k, was chosen to be 3. Since the goal was to capture the correlation among the predictions, a correlation based distance was used.

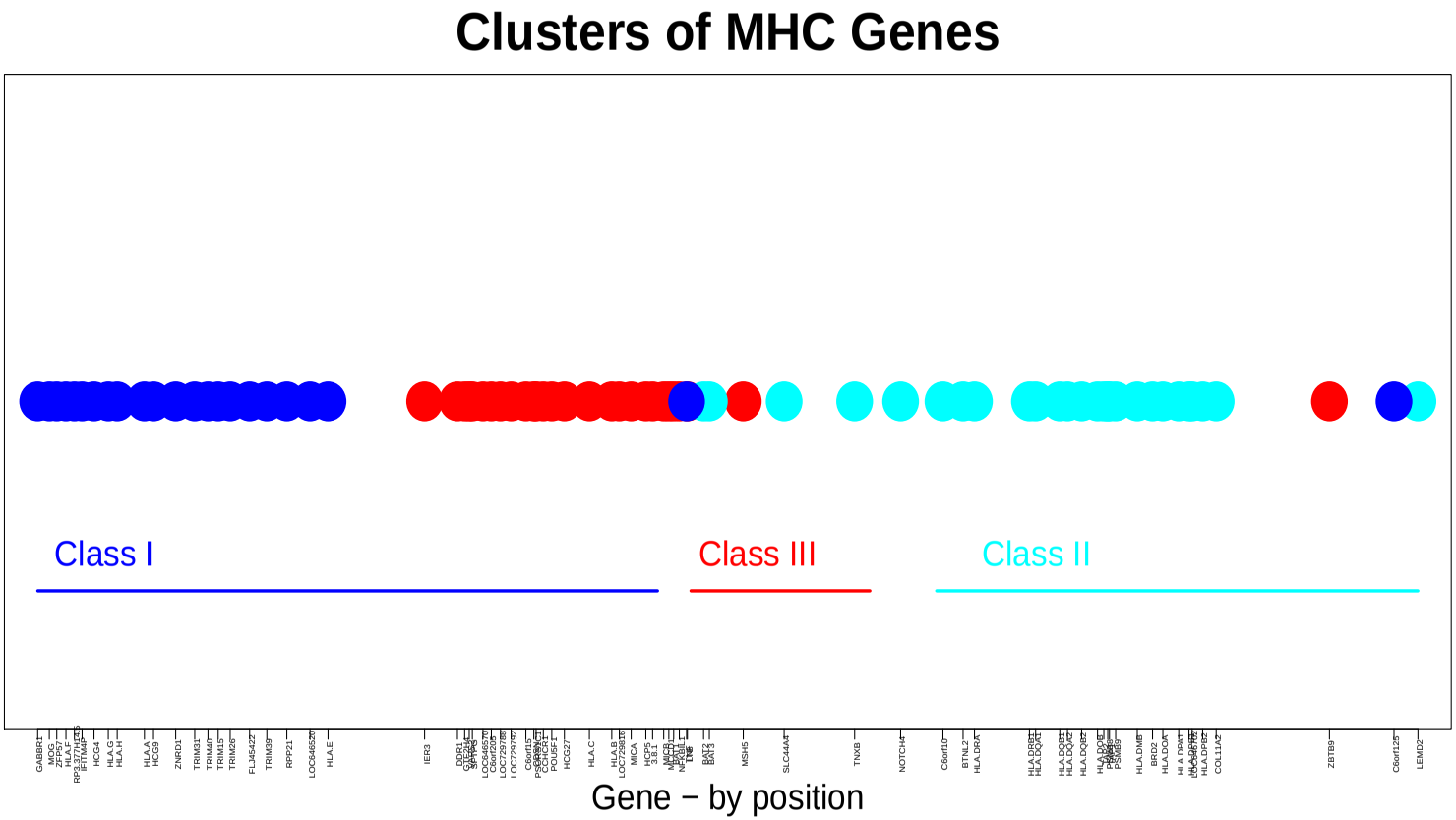


Figure 3.8: Clusters of the MHC genes based on the $\hat{f}(X)$ for cases. Genes are organized by position on Chromosome 6. The lines underneath indicate the boundaries for the three MHC classes. The three clusters of genes correspond very closely with their location within chromosome 6 and the cluster memberships correspond almost perfectly with the three MHC classes.

Figure 3.8 shows the clusters of the 78 MHC genes. The genes, are ordered by their position on Chromosome 6. The groupings of genes clearly correspond to their position, suggesting that the correlation between the $\hat{f}(X)$ is maintained through the position. Of greater interest, the cluster memberships correspond fairly well with the three MHC classes.

To explore the question of whether these clusters represented independent signals, a SuperLearner was trained using the SNPs in each of the three clusters, using the same procedures of above. The interest was not in whether each cluster is associated with MS (each clearly is), but whether the associations represent distinct signals. Table 3.4 shows the correlation matrix for the three clusters. While there is positive correlation between all three classes, Class I and Class II genes appear to be fairly independent of one another compared to Class III (which is physically located in between Class I & II). This suggests that there may be two separate associations within the MHC for MS. While this analysis does not represent a confirmation of independent signals, like previous analyses, it does correspond to previous analyses that they may exist.

	Class I	Class II	Class III
Class I	1.000	0.175	0.410
Class II	0.175	1.000	0.516
Class III	0.410	0.516	1.000

Table 3.4: Correlation matrix of the $\hat{f}(X)$ from the three MHC clusters, corresponding to the three MHC classes. Class I & II appear to be somewhat independent, suggesting that there may be two independent signals within the MHC for MS.

3.6.3 Thoughts on Application to Genetic Data

Three different applications to genetic data were illustrated, highlighting the flexibility of this approach. One particular challenge to this method is the ability to create a strong predictor using genetic data. The influence of ones genetics on disease (i.e. $P(\text{Disease}|\text{Genetics})$) is going to be relatively low. For example, while MS has a strong genetic component, its overall heritability is only estimated at 25% [Oksenberg and Barcellos, 2005]. Therefore, if a gene is in fact causative of disease, one would not expect the $P(D|G)$ to differ much from average risk. Therefore the true risk (θ) will not be much less than the expected risk (θ^*) making detecting a significant association challenging.

Clayton [2009] looked at SNP data and noted that even highly associated SNPs have low predictive ability. However, Kooperberg et al. [2010] recently showed, that creating prediction models using SNPs that do not have strong marginal associations,

does improve the models. This is essentially the approach undertaken here, using all SNPs within a region regardless of their marginal association. Even so, it is possible that the weak associations detected in the candidate gene analysis, may not be indicative of a lack of effect, but simply a weak predictive ability of the SNPs in those genes.

3.7 Conclusion

The proposed method represents a powerful, flexible and semi-parametric approach to testing the relationship between a set of variables and an outcome. It has applicability both within the fields of prediction and machine learning as well as classical statistical inference. From a machine learning perspective, it represents a means of assessing whether the predictive ability of a set of predictors is better than what would be expected by chance. This represents an important but often unasked question of whether one should even attempt to construct a predictor from the given data. From an inferential perspective, it represents a means to assess whether a set of variables is related to an outcome. Traditional methodology is aimed at assessing the relationship between one covariate and one outcome. This allows one to look at multiple variables at once. This is particularly important in fields like genetics and the social sciences where one often has data on one level (e.g. SNPs, social variables) and wants to make inference on another level that aggregates the data (i.e. genes, SES). Such aggregation is often the only approach, as many times the level one wants to make inference on, represents a construct and not an actual observable variable.

The constructed test statistic is a Wald statistic, using influence curves to calculate the inference. A finite sample correction was necessary to appropriately scale the variance. While the current correction is somewhat adhoc it conforms with other current work in the area. An important area of further investigation is a formalized correction. For perspective, the test-statistic has been compared to the F-test used in linear regression. In this sense it can be thought of generalized goodness-of-fit test. Simulation results show the relationship to the F-test, but also how it is more powerful under a variety of alternatives.

A range of applications to genetic data were illustrated. The first involved testing the association of gene (and pathway) with disease. A more comprehensive simulation illustrated how this is more powerful than typical approaches under complex associations. While the candidate gene analysis was not able to reveal associations greater than what would be expected by chance, examination of results did reveal that genes that would not be identified by traditional approaches had strong associations. An analysis of genetic pathways was able to more strongly confirm results in a previous study that were merely speculated upon. A final analyses illustrated the variety of questions that could be addressed with this data. Using the same dataset, the goal was to find clusters of genes in the MHC. The clusters corresponded almost perfectly

with biological understanding and the results provided insight that there may be two independent sources of association within the MHC for MS.

In all, this method represents an important contribution to a range of statistical applications, filling a need within both statistical learning and inferential statistics. It effectively expands the range of questions that one can ask of their data and should represent an important tool for many analyses.

Bibliography

Y. Bengio and Y. Grandvalet. No unbiased estimator of the variance of k-fold cross-validation. *Journal of Machine Learning Research*, 5:1089–1105, 2004.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57: 289–300, 1995.

J Beyene, D. Tritchler, J. L. Asimit, and J.S. Hamid. Gene- or region-based analysis of genome-wide association studies. *Genetic Epidemiology*, 33:s105–s110, 2009.

M. D. Birkner, A. E. Hubbard, and M. L. van der Laan. Data adaptive pathway testing. Technical Report 197, U.C. Berkeley Division of Biostatistics, November 2005.

L. Breiman. Stacked regressions. *Machine Learning*, 24:49–64, 1996.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

F.B. Briggs, B.A. Goldstein, J.L. McCauley, R.L. Zuvich, P.L. De Jager, J.D. Rioux, A.J. Ivinson, A. Compston, D.A. Hafler, S.L. Hauser, J.R. Oksenberg, S.J. Sawcer, M.A. Pericak-Vance, J.L. Haines, L.F. Barcellos, and International Multiple Sclerosis Genetics Consortium. Variation within dna repair pathway genes and risk of multiple sclerosis. *American Journal of Epidemiology*, 172:217–224, 2010.

P. Chaffee, A. E. Hubbard, and M. L. van der Laan. Permutation-based pathway testing using the super learner algorithm. Technical Report 263, U.C. Berkeley Division of Biostatistics, March 2010.

D.G. Clayton. Prediction and interaction in complex diseases. *Plos Genetics*, 5, 2009.

B. A. Cree, J. D. Rioux, J. L. McCauley, P. A. Gourraud, P. Goyette, J. McElroy, P. De Jager, A. Santaniello, T. J. Vyse, P. K. Gregersen, D. Mirel, D. A. Hafler, J. L. Haines, M. A. Pericak-Vance, A. Compston, S. J. Sawcer, J. R. Oksenberg, S. L. Hauser, IMAGEN, and IMSGC. A major histocompatibility Class I locus contributes to multiple sclerosis susceptibility independently from HLA-DRB1*15:01. *PLoS ONE*, 5:e11296, 2010.

T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

S. Dudoit and M. J. van der Laan. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodology*, 2:131–154, 2005.

R.A. Fisher. Questions and answers no.14. *The American Statistician*, 2:30–31, 1948.

Y. Grandvalet and Y. Bengio. Hypothesis testing for cross-validation. Technical Report 1285, Departement dInformatique et Recherche Operationnelle, August 2006.

D. A. Hafler, A. Compston, S. Sawcer, E. S. Lander, M. J. Daly, P. L. De Jager, P. I. de Bakker, S. B. Gabriel, D. B. Mirel, A. J. Ivinson, M. A. Pericak-Vance, S. G. Gregory, J. D. Rioux, J. L. McCauley, J. L. Haines, L. F. Barcellos, B. Cree, J. R. Oksenberg, and S. L. Hauser. Risk alleles for multiple sclerosis identified by a genomewide study. *N. Engl. J. Med.*, 357:851–862, Aug 2007.

T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning*. Springer, New York, 2 edition, 2009.

A. G. Heidema, J. M. Boer, N. Nagelkerke, E. C. Mariman, D. L. van der A, and E. J. Feskens. The challenge for genetic epidemiologists: how to analyze large numbers of SNPs in relation to complex diseases. *BMC Genetics*, 7:23, 2006.

IMSGC. Comprehensive follow-up of the first genome-wide association study of multiple sclerosis identifies kif21b and tmem39a as susceptibility loci. *Human Molecular Genetics*, 19:953–962, 2010.

C. Kooperberg, M. Leblanc, and V. Obenchain. Risk prediction using genome-wide association studies. *Genetic Epidemiology*, pages 643–652, 2010.

S. Lee. Mistakes in validating the accuracy of a prediction classifier in high-dimensional but small-sample microarray data. *Statistical Methods in Medical Research*, 26:1102–1113, 2007.

L. Lusa, L.M. McShane, M.D. Radmacher, J.H. Shih, G.W. Wright, and R. Simon. Appropriateness of some resampling-based inference procedures for assessing performance of prognostic classifiers derived from microarray data. *Statistics in Medicine*, 26:1102–1113, 2007.

B.M. Neale and P.C. Sham. The future of association studies: gene-based analysis and replication. *American Journal of Human Genetics*, 75:353–362, 2004.

J. R. Oksenberg and L. F. Barcellos. Multiple sclerosis genetics: leaving no stone unturned. *Genes and Immunity*, 6:375–387, Aug 2005.

S. Purcell, B. Neale, K. Todd-Brown, L. Thomas, M. A. Ferreira, D. Bender, J. Maller, P. Sklar, P. I. de Bakker, M. J. Daly, and P. C. Sham. PLINK: a tool set for whole-genome association and population-based linkage analyses. *Am. J. Hum. Genet.*, 81:559–575, Sep 2007.

M.D. Radmacher, L.M. McShane, and Simon R. A paradigm for class prediction using gene expression profiles. *Journal of Computational Biology*, 9:505–511, 2002.

K.M. Ting and I.H. Witten. Stacked generalization: when does it work? In *Procs. International Joint Conference on Artificial Intelligence*, pages 866–871, 1997.

M. J. van der Laan, S. Dudoit, and A.W. van der Vaart. The cross-validated adaptive epsilon-net estimator. *Statistics & Decisions*, 3:373–395, 2006.

M. J. van der Laan, E. C. Polley, and A. E. Hubbard. Superlearner. *Statistical Applications in Genetics & Molecular Biology*, 6, 2007.

M.J. van der Laan. Statistical inference for variable importance. *The International Journal of Biostatistics*, 2, 2006.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.

WTCCC. Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature*, 447:661–678, 2007.

Chapter 4

A Direct Approach to Analyzing Gene Expression Data

4.1 Introduction

In many gene expression experiments a large number of genes will prove to be differentially expressed even after controlling for multiple testing. Secondary analyses such as Gene Ontology (GO) and Motif Analysis (MA) have the potential to provide biological insight and direct future studies, however the methods need to be used thoughtfully. Simply, running all the significant genes through an analysis program without proper filtering may lead to muddled results. Therefore a central statistical challenge is how to group genes to be passed on to a secondary analysis.

Eisen et al. [1998] was one of the first papers to address this issue, showing that clustering data could lead to more homogeneous groupings of the data and more meaningful results. In recent years highly sophisticated clustering methods have been developed for gene expression studies (e.g. see van der Laan and Pollard [2003]). Clustering is a very effective tool for grouping data and has become standard practice in gene expression studies. After clusters are identified, they are passed on to the appropriate secondary analysis for further investigation [Werner, 2001].

While this approach has led to many valuable discoveries it ultimately does not make full use of the data. In many gene expression experiments a phenotype pattern is observed that one desires to explain. In our recent study of *C. elegans* [Chen et al., submitted], four different mutants were created, along with a wild type (Wt), and observed under dietary restriction. The mutants consisted of a knockout of the Daf-2 gene (D2), a knock-out of the S6k gene (S6k), a knock-out of both genes (DM), and a triple mutant that also knocked out the Daf-16 gene (TM). Observation of the different groups showed that mutant type had a significant impact on lifespan (see figure 4.1). It was expected that knocking out both the S6k gene and Daf-2 genes would increase lifespan [Chen et al., submitted]. Likewise, it was not a surprise that

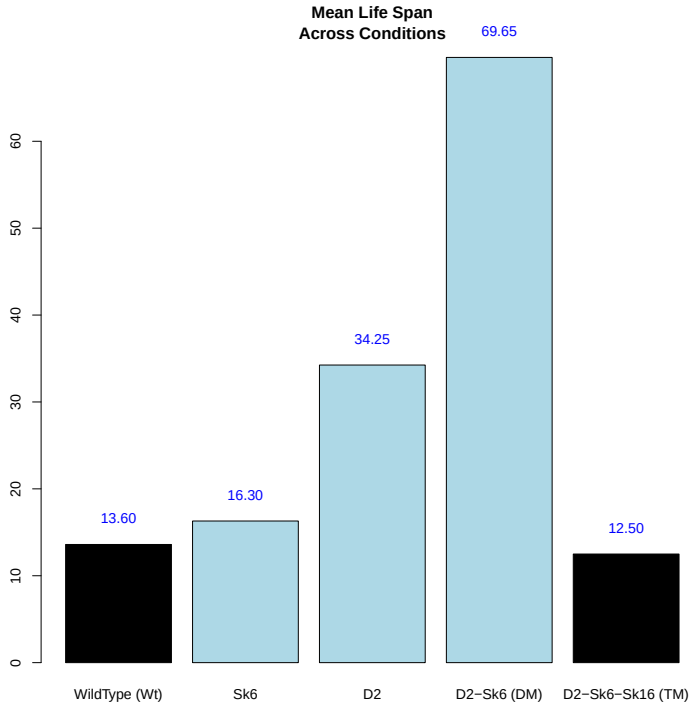


Figure 4.1: Observed mean lifespan of the different mutant types. The non-linear increase in lifespan in the double mutant was an indication that there is something of interest occurring when both the S6k and Daf-2 genes are suppressed.

knocking-out the Daf-16 gene would arrest this process. However, what was of great interest was the apparent interaction between the S6k and Daf-2 knockout, based on the greater than 5-fold increase of lifespan in DM.

Generally, suppressing a gene produces a cascade effect that will influence the expression of other genes. The goal of the knockout experiment is not to determine whether the knocked-out gene effects the phenotype, but which other genes are also altered - either by being over or under expressed. This can suggest a biological pathway for expression of the phenotype of interest. Observation of the lifespan data led to the hypothesis that there is an interaction between the Daf-2 and S6k gene that led to an increase in lifespan. Two biological hypotheses are put forward. The first is that suppression of both genes led to an increase in the change of expression of other genes. This is termed the *quantitative* interaction hypothesis. The second, and in some ways more compelling hypothesis, is that suppressing the two genes leads to a previously unseen change in some other genes. This is dubbed the *synergistic* or *qualitative* interaction hypothesis.

Given these results, the goal is to determine which genes are being changed in the

double mutant case and whether any biological insight could be gleamed from them. To accomplish this, both GO and MA analysis were undertaken. GO analysis is a means to determine whether a set of genes possess a common function more than what would be expected by chance. MA examines the sequences of different genes to see if they share certain patterns. Such patterns, or motifs, might be indicative that they share a common transcription or regulatory factor.

In analyzing the data, one would first perform a statistical test to determine which genes are differentially expressed in the double mutant condition. After, it would be necessary to appropriately group genes to pass on to the secondary GO and MA programs. This would typically be accomplished via clustering. Given the nature of the experiment, and the finer hypotheses, a more targeted approach was undertaken to explicitly match genes to different hypotheses. Such an analysis results in a very homogeneous group of genes, more than would be accomplished via clustering. After passing these groups to the secondary analysis significant biological discoveries were made that were able to be replicated and contribute to the understanding of the genetic mechanism behind lifespan in *C. elegans*.

4.2 Methods

4.2.1 Data

See Chen et al. [submitted] for discussion of the biological experiment. Briefly, a wild type and four mutant *C. elegans* were created. The four mutants consisted of a knockout of the Daf-2 gene (D2), a knockout of the S6k gene (S6k), a knockout of both genes (DM), and a triple knockout (TM) of the Daf-16 gene in addition to Daf-2 and S6k. TM was meant to serve as a secondary control. For each of the five groups, 10 *C. elegans* were tested for a sample of 50 specimens. Life span was monitored in each specimen with average lifespan shown in figure 4.1.

Genome wide gene expression data was assessed with the Roche NimbleGen microarray. There are $119,657$ probes across $23,945$ genes on the array, with most genes having 5 probes. The different probes potentially represent different isoforms or splices of the gene and are therefore each biologically meaningful.

4.2.2 Hypotheses

Based on the observed lifespan data two primary hypotheses were generated.

1. A probe is only differentially expressed with the suppression of both the Daf-2 and S6k genes. This is the *synergistic* or *qualitative* interaction hypothesis.

2. A probe is differentially expressed to a greater extent within the double mutant. This is the *quantitative* interaction hypothesis. Within this hypothesis there are three sub hypotheses

(a) A probe is differentially expressed in D2 and the expression is amplified in DM

(b) A probe is differentially expressed in S6k and the expression is amplified in DM

(c) A probe is differentially expressed in D2 both S6k and the expression is amplified in DM

It should be noted that there are some scenarios excluded from these hypothesis. For example, if a probe is differentially expressed in D2 and then suppressed in DM this would not be covered. It is this specific hypothesis that gives the following analysis its strength. Since the goal of the analysis is to explain the observed 5.5 fold increase lifespan for DM, we are looking for probes that are exaggerated in DM.

4.2.3 Primary Analysis

Linear Model

Lowess normalization using the limma package in R was used to normalize the expression data [Smyth, 2005]. A general linear model was fit to each probe (Y_i) also using the limma package. The model was

$$E(Y_i|X) = \beta_1 I(Baseline) + \beta_2 I(D2) + \beta_3 I(S6k) + \beta_4 I(D2) * I(S6k) \qquad (4.1)$$

Since each term is a series of indicators, this represents a non-parametric model of the test of the E[expression|mutant type]. The baseline was used to be both the Wt and the TM specimens.

The linear model was used to examine each of the four hypotheses. The primary parameter of interest is β_4. This represents the interaction in the double mutant. A rejection of the null hypothesis that $\beta_4 = 0$ for a given probe, indicates that in the double mutant there is a departure from additivity in the expression of that probe in the double mutant.

As indicated in the hypothesis section, not all scenarios where $\beta_4 \neq 0$ are of interest. Following the example from above, if $\beta_2 > 0$, $\beta_3 = 0$ and $\beta_4 < 0$, this would not be of biologic interest since the DM does not lead to an amplification of expression of the particular probe. Table 4.1 shows the beta values for the corresponding hypothesis.

Selection of Probes

The linear model was fit first using WT as the baseline. To estimate the standard errors an empirical Bayes procedure was used, also through the limma package [Smyth, 2004]. The empirical Bayes approach calculates a moderated t-statistic for each contrast of interest. The parameters to calculate the posterior variance of the

Hypothesis	β **Value**
1	$\beta_2 = \beta_3 = 0,\ \beta_4 \neq 0$
2 (a)	$\beta_3 = 0,\ \beta_2, \beta_4 \neq 0,\ sign(\beta_2) = sign(\beta_4)$
2 (b)	$\beta_2 = 0,\ \beta_3, \beta_4 \neq 0,\ sign(\beta_3) = sign(\beta_4)$
2 (c)	$\beta_2, \beta_3, \beta_4 \neq 0,\ sign(\beta_2) = sign(\beta_3) = sign(\beta_4)$

Table 4.1: Table of hypotheses and the corresponding β values

t-statistic are based off the observed the data. For gene expression studies that often have many probes to test, but few samples, this provides an efficient means to estimate the inference for a test statistic. The posterior variance is estimated as:

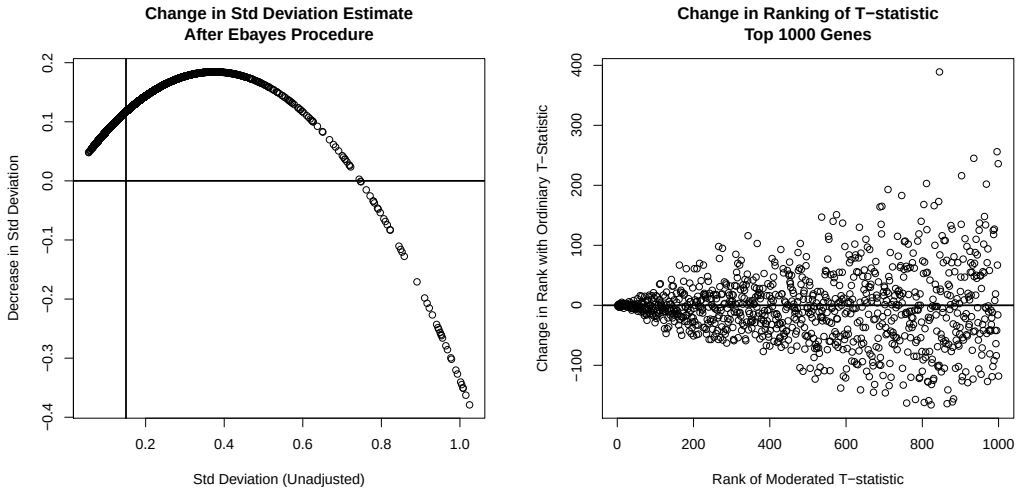
$$\tilde{s}_j^2 = \frac{d_0 s_0^2 + d_j s_j^2}{d_0 + d_j} \tag{4.2}$$

This represents the convex combination of the prior variance (s_0^2) with the observed variance (s_j^2), scaled by the degrees of freedom (d_0 & d_j respectively). An inverse Chi-square distribution is used for the prior, which is the typical conjugate prior.

The empirical Bayes procedure results in an non-monotonic change in the test statistics. This can create a more robust statistic[1]. Given the modest sample size and the fact that there is not a perfectly robust procedure for deriving inference at this sample size (i.e. no obvious permutation test and n is too small to rely on a bootstrap) one must parametrically estimate the variance. The empirical Bayes procedure shrinks large estimates of the $var(\hat{\beta}_{ij})$ (i.e. s_0^2) towards more "typical" values. The induced shrinkage will most significantly impact those genes with larger variances (see Figure 4.2(a)). This will often have a conservative outcome as genes that would have been selected only because s_0^2 is relatively small (and not $|\hat{\beta}_j|$ large) become relatively insignificant. This will ultimately lead to a different ranking of the genes (see Figure 4.2(b)).

The p-values for each of the contrasts of interest were adjusted to control the false discovery rate (FDR) using the Benjamini-Hochberg (BH) procedure [Benjamini and Hochberg, 1995]. For each of the four hypotheses, probes were selected where the corresponding β values had an adjusted p-value less than 0.05. Due to the presence of a second baseline group (TM) it was possible to add robustness to the final probe set. The analysis was repeated using TM as the baseline. Only those probes that matched a hypothesis in both analyses were passed onto the secondary analyses.

[1]Moreover, because $\tilde{s}_j^2 \xrightarrow[n\to\infty]{} s_0^2$ any bias introduced will go away asymptotically.



(a) Change Std Deviation (b) Change Rank

Figure 4.2: Impact of the empirical Bayes procedure on test statistics. Figure (a) shows the change in the standard deviation estimate for a random set of 10,000 genes. Most values are decreased (indicated by being above the 0 line), with those with smallest empirical standard deviations receiving less shrinkage. The vertical line indicates the mean of the unadjusted standard deviation, illustrating that the greatest shrinkage occurs for larger values. For extreme values, the standard deviation is actually increased due to the lack of prior mass for the empirical Bayes procedure. This leads to an overall change in the ranking of the genes (b). The top 1000 genes based on the moderated t-statistic are compared to the rank without moderation. The funnel shape indicates that the change in ranking becomes more dramatic as one goes further down the list.

4.2.4 Secondary Analysis

Secondary analysis is an important part of gene expression studies, as they allow one to gain further insight into the type of changes that occur in the experiment. They can illuminate both the function of the genes that are differentially expressed and show how they may be regulated. While most secondary analysis programs will provide a p-value with the output these should not be thought of as true hypothesis testing. Instead, these are qualitative analyses where the p-values should be interpreted as a ranking of top findings. In the current study, two secondary analyses were performed: MA and GO.

Motif Analysis

MA is a process of finding patterns in the sequences of different genes. If a group of genes share a sequence pattern upstream from the gene, it is possible that those genes are regulated by the same transcription factor (TF). Identifying such a TF could provide a means of controlling the expression of such a group of genes.

There are number of different approaches for discovering motifs utilizing a range of statistical methods including the EM algorithm, Gibbs sampling and genetic algorithms [Das and Dai, 2007]. While there are number of available programs for MA, the Amadeus platform was used [Linhart et al., 2008]. In Amadeus, motifs are *discovered* through a series of phases, where in each phase the requirements for passing on a motif are refined. The p-value is based on determining the probability of seeing that motif occurrence, under the null hypothesis that the genes in the target set were drawn randomly, independently, and without replacement from the background set. This is calculated via the hypergeometric distribution. Unlike in GO analysis (see below) there is no notion of how many different motifs are tested so there is no standard control for multiple testing[2].

Like most MA software, Amadeus allows the user to input a list of genes for a specific organism. After specifying a region to search and a motif length, overrepresented motifs are identified. One advantage of Amadeus is that it then compares the identified motif to a list of known TF motifs, based on the TRANSFAC database and any similar TF motifs are identified. Of course, not all identified motifs will have a corresponding TF binding site while some may have multiple similar one.

For the MA, the unique genes were divided into those of being over or under expressed. The unique genes for each of the eight sets were fed into Amadeus. Motifs of length 8 and 10 base-pairs (bp) were searched for in a range of 1500 bp upstream of the gene to 400 bp downstream.

[2]Amadeus does allow the user to control for multiple testing via a bootstrap procedure. However, since the goal of MA is qualitative in nature, and the bootstrapping can be very computational it was deemed unnecessary

Gene Ontology Analysis

GO analysis is a means of determining whether groups of genes share a similar function or *ontology*. Known genes are placed on a directed acyclic graph (DAG) based on their biological, molecular or cellular functions. Given a collection of genes and a reference set, an analysis is performed whether any of the terms are over-represented, typically by comparing to the hypergeometric distribution, in what is referred to as the *term-for-term* approach.

Using the notation from Grossmann et al. [2007], to calculate the statistic we assume that there is a population of genes, P and study set S, of size m and n respectively. The term one is interested in is denoted by t and P_t and S_t refer to those genes with those terms in the population and study sets, each being size m_t and n_t. Now, let a set Σ of size n be randomly sampled without replacement from P, and let σ_t be the number of genes in Σ with term t. The probability of observing σ_t annotations can be calculated via the hypergeometric distribution

$$P(\sigma_t = k) = \frac{\binom{m_t}{k}\binom{m-m_t}{n-k}}{\binom{m}{n}} \tag{4.3}$$

To assess the probability, one sums (4.3) from n_t to the maximum possible number of annotations.

$$P(\sigma_t \geq n_t) = \sum_{k=n_t}^{min(m_t,n)} \frac{\binom{m_t}{k}\binom{m-m_t}{n-k}}{\binom{m}{n}} \tag{4.4}$$

This presents a one-sided Fisher exact test.

Grossmann et al. [2007] showed that there is actually a flaw in this test statistic since it does not take into account dependencies between parent and child terms. They proposed another statistic, which they termed the *parent-child* approach.

Let $pa(t)$ denote the parents of t. Based on the hierarchical tree $m_{pa(t)} \geq m_t$.

$$P(\sigma_t = k | \sigma_{pa(t)} = n_{pa(t)}) = \frac{\binom{m_t}{k}\binom{m_{pa(t)}-m_t}{n_{pa(t)}-k}}{\binom{m_{pa(t)}}{n_{pa(t)}}} \tag{4.5}$$

P-values are calculated as in (4.4). Grossmann et al. showed that this approach gives more stable results and is implemented in the program Ontologizer.

While the results are inherently qualitative, they provide a valuable degree of context to a gene expression experiment and can present new directions for research, if unexpected terms appear. GO analysis was performed for the probe sets for each of the 4 hypotheses. Significant terms were discovered using the parent-child analysis using an FDR cutoff of .10.

4.2.5 Follow-up Analysis

Based on the MA and GO analysis targets of interest were determined for follow-up analysis by the consulting biologist. For the MA, since it is more straightforward to suppress a TF than to induce one, genes that control the TFs that showed indications of being up-regulated were targeted. *C. elegans* with the TF knocked out were created for each of the four conditions and lifespan was monitored. Based on GO analysis, gene groups of interest were also followed up of for further suppression studies and analysis, though have not yet been performed.

4.2.6 Comparison to Standard Analysis

To compare the results to standard practice, a cluster analysis was performed. The clustering algorithm chosen was HOPACH [van der Laan and Pollard, 2003]. HOPACH is a flexible clustering algorithm well suited for working with large genetic data. HOPACH, combines the strength of both centroid based algorithms (e.g. k-means, PAM) and hierarchical approaches (e.g. agglomerative, divisive). Like hierarchical algorithms, one does not need to pre-specify the number of clusters in the data. This is desirable in genetic data as the number of clusters is often unknown. However, like centroid based algorithms, one receives a strong notion of cluster membership, which is important for grouping of the data. The HOPACH algorithm is implemented in the R package hopach.

To select probes, the same linear model was fit, with β_4 being the only parameter of interest. Probes that showed significant changes in β_4 with respect to both the WT and TM were selected for clustering. The set of probes were clustered using the cosine angle distance metric. Since clustering using hierarchical methods often results in a few large clusters and many small or singleton clusters, only the large clusters were examined. To compare the quality of each method for grouping the data, the silhouette distances [Rousseeuw, 1987] based on correlation distance were calculated for each approach. MA and GO analysis were performed for the HOPACH based clusters and the results were compared.

4.3 Results

4.3.1 Probe Selection

Gene expression data was successfully collected on 47 of the 50 samples (94%). The linear model in equation (4.1) was fit to the data using the Wt and TM as baseline. All probes that had an BH-FDR adjusted p-value less than .05 on β_4 were selected. There were $20,563$ and $32,854$ such probes between the WT and TM baseline groups respectively. Of those, $12,406$ were overlapping and were passed on to the comparative analysis (see below).

Groups of probes were selected based on the hypotheses define in Table 4.1. Table 4.2 lists the number of significant probes probes per hypothesis, including the number overlapping between the two conditions. The largest grouping were those from Hyp1 and Hyp2a.

	Wt Baseline	TM Baseline	Overlap
Hyp 1	3386	5558	1444
Hyp 2a	1911	1798	806
Hyp 2b	300	54	6
Hyp 2c	262	130	42

Table 4.2: Number of Significant Probes from each analysis

4.3.2 Motif Analysis

The four probe sets were divided into up and down regulated groups and MA was performed. Due to sample sizes only the probes from Hyp1 and Hyp2a were analyzed (Hyp2b and Hyp2c had too few probes to obtain meaningful results). MA was run for each probe set twice - looking for motifs of length 10 and 8 respectively. The top five motifs were inspected for any transcription factors of interest, determined by the consulting biologist.

Hypothesis (Direction)	Motif Length	Motif P-value (Rank)	TF of Interest
Hyp 1 (Up)	10	3.3×10^{-18} (5)	*HSF*
Hyp 1 (Up)	8	2.3×10^{-19} (1)	*COUP-TF:HNF-4, XBP-1, ABF, HTF, EmBPm, HNF-4alpha, NMyc, c-Myc*
Hyp 1 (Up)	8	9.0×10^{-13} (3)	*ABF1*
Hyp 2a (Up)	10	4.4×10^{-17} (5)	*FOX03, FOX04, DAf-16, FOX01*
Hyp 2a (Up)	8	3.6×10^{-12} (2)	*RP58*
Hyp 2a (Up)	8	4.2×10^{-12} (3)	*HSF*
Hyp 1 (Down)	10	9.0×10^{-13} (2)	*YY1*
Hyp 1 (Down)	8	9.0×10^{-13} (1)	*SREBP1*
Hyp 2a (Down)	10	4.4×10^{-17} (3)	*XFD-3*
Hyp 2a (Down)	8	3.6×10^{-12} (2)	*SREBP-1, ROAZ*

Table 4.3: Listing of the *interesting* MA results as determined by the consulting biologist.

Figure 4.3: GO Tree for Hyp 1 condition. Significant terms are highlighted.

4.3.3 Go Analysis

GO analysis was performed on the four hypothesis gene sets. Significant GO terms were selected with a BH-FDR adjusted p-value less than .10. Again, due to set sizes, the only two analyses with significant terms were Hyp1 and Hyp2a. The GO trees are shown in figures 4.3 and 4.4. Due to the complexity of the GO tree for Hyp2a, figure 4.5 lists the significant terms.

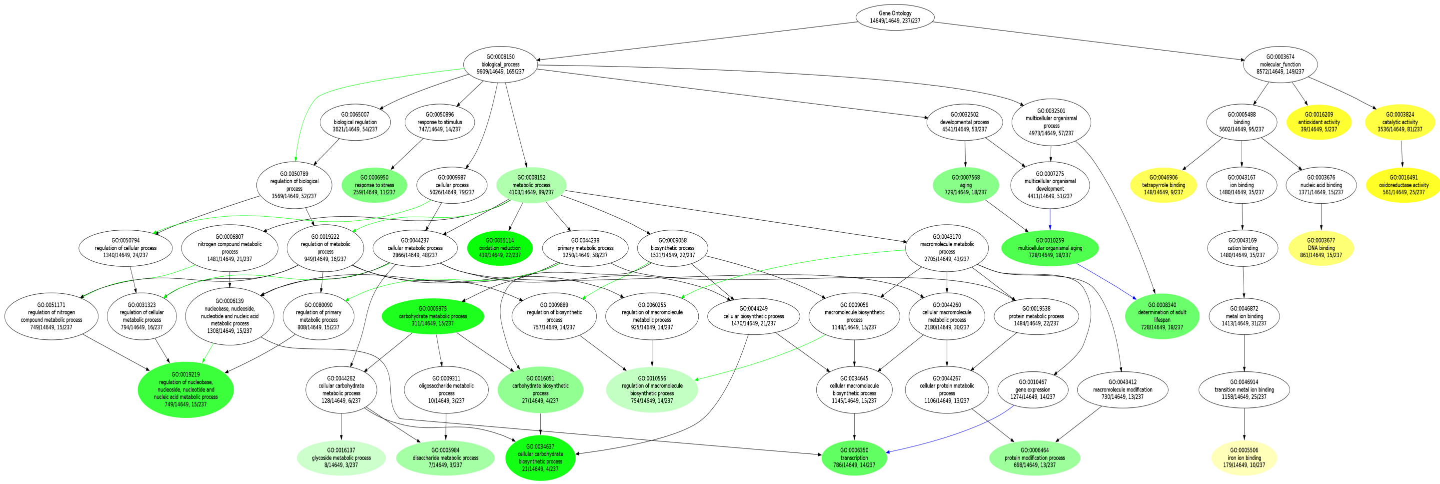


Figure 4.4: GO Tree for Hyp 2a condition. Significant terms are highlighted.

The Hyp2a GO analysis revealed a number of interesting findings. Not surprisingly, but a good confirmation, genes associated with aging and lifespan were overrepresented (figures 4.4 & 4.5). Of greater interest, terms reflecting phosphorylation (figure 4.3) and carbohydrate processes (figures 4.4 & 4.5) were represented, providing additional hypotheses for the mechanism behind life span extension.

4.3.4 Follow-up Analysis

Since experimentally, it is more straightforward to follow-up on upregulated TFs (it is easier to suppress a TF than induce) only those have been examined to date. Nineteen genes corresponding to the TF targets of interest were identified by the consulting biologist. If the TF is involved in increased lifespan observed in the mutants, than suppression should return lifespan back to baseline. Table 4.4 shows the average lifespan under different mutant conditions. The Mack-Skilling statistic [Mack and Skilling, 1980] was calculated comparing each of the TF knockouts to the control, and the associated p-value is shown. The Mack-Skilling test is a non-parametric test comparing k treatments across b blocks. In the present study, each knockout was compared independently against the control case so $k = 2$. The b blocks were the 4 mutant classes. The statistic was implemented in R and calculated as described in Hollander and Wolfe [1999]. The χ^2 approximation was used to calculate the p-value.

Unsurprisingly Daf-16 showed the greatest degree of suppression (this is why it was used as a control in the initial experiment). Of the other 18 genes, 11 showed indication of being involved in lifespan extension. Of greatest interest were those that showed a suppression in the DM. Follow-up Wilcoxon tests were performed just comparing the DM in the control and knockout groups. Eight of the 19 had Bonferonni adjusted p-values less than 0.05. These were: *daf-16*, *hsf-1*, *xbp-1*, *F17A9-3*, *mxl-3*, *che-1*, *nhr-64*, *hlh-10*. These genes represent further targets for study.

The GO analysis suggested potential direction for follow-up. The Hyp1 analysis suggested phosphorylation was overrepresented (figure 4.3) and biochemical analysis indicated that the s6k single mutant and DM have increased phosphorylation levels of AMPK. This is essential for the synergistic lifespan extension by DM [Chen et al., submitted]. It is speculated that there might be additional kinases and/or phosphatases involved in this regulation. Therefore, examination of the genes from the DM over-represented group with the GO term "macromolecule modification/phosphorus metabolic process" will be further analyzed for their effects on AMPK phosphorylation and DM synergistic lifespan extension. Similarly, there are biochemical explanations for why carbohydrate processes may be involved in lifespan extension and these too will be further analyzed.

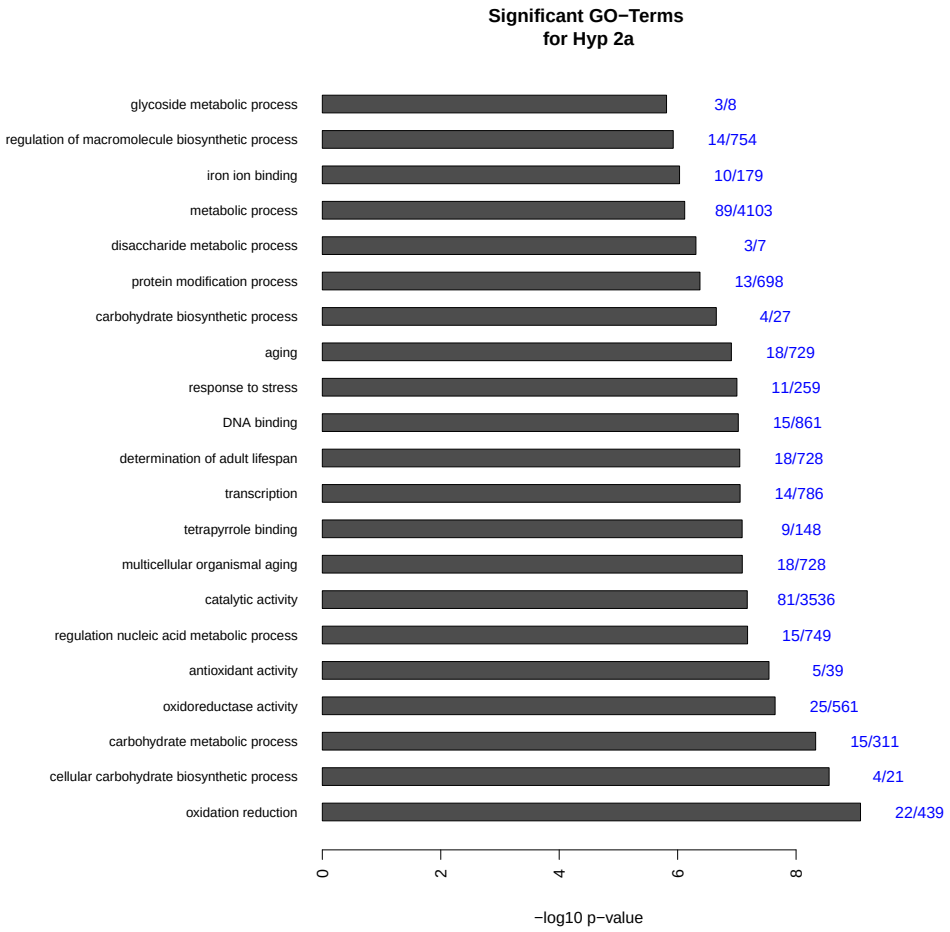


Figure 4.5: GO Terms for Hyp 2a condition. Bars reflect $-log_{10}$ of the p-values. Labels at end of bars indicate the number of probes in the hypothesis set versus the number of probes in the full dataset had the term.

4.3.5 Comparative Analysis

To compare the modeling approach to a more standard analysis, the data was also clustered, as would be done in a typical micro-array experiment. There were 12,406 probes that showed differential expression in β_4 with respect to both WT and TM. After clustering, there were 668 clusters, five of which had more than 500 members. The silhouette distance for these five clusters are shown in figure 4.6. The Hyp1 and Hyp2a groupings were divided into up and down regulated groups and the corresponding silhouette are shown in figure 4.7.

The results suggest that the model based groupings are a bit tighter and more homogeneous. It is noted that there is some negative silhouettes for Hyp1. This is not surprising since the silhouette was based on correlation and one would expect that

Gene Knock-Out	N2	S6k	D2	DM	MS p-value
control	12.5	14.5	23.0	52.4	NA
daf-16	9.0	8.9	9.0	11.6	$< 1 \times 10^{-16}$
hsf-1	7.5	6.2	9.8	12.0	$< 1 \times 10^{-16}$
xbp-1	11.6	10.8	20.7	36.7	1.1×10^{-16}
aha-1	13.2	14.9	22.0	46.0	0.86
F17A9.3	11.9	15.9	22.1	43.7	0.27
sbp-1	10.4	9.8	21.5	50.3	1.1×10^{-16}
nhr-85	11.4	13.0	28.2	48.3	0.013
hlh-30	12.5	12.2	15.5	43.2	3.9×10^{-14}
pha-4	10.1	12.3	24.6	48.9	2.1×10^{-8}
mxl-3	13.2	14.9	23.4	39.2	0.50
che-1	13.5	14.7	23.5	37.9	0.71
alr-1	11.6	13.8	23.9	44.0	9.8×10^{-4}
unc-55	12.7	14.2	22.3	48.8	0.25
ahr-1	11.8	15.1	23.1	47.5	0.19
nhr-64	12.9	15.0	23.6	42.2	0.73
rnt-1	12.3	13.7	20.3	45.9	5.7×10^{-4}
crh-1	11.9	11.3	21.2	49.8	1.8×10^{-9}
pag-3	12.3	13.9	21.2	44.9	0.0010
hlh-10	12.0	14.4	23.8	40.4	0.0069

Table 4.4: Average Lifespan for the control and various knock-outs. Genes where suppression lead to the greatest departure from control lifespan are highlighted.

for Hyp1 the expression values for the D2 and S6 conditions would be more random than what one would see in Hyp2. Overall, it is noteworthy, that an approach that was not aimed explicitly at detecting homogeneous groupings was more capable of doing so than an approach that was attempting to find homogeneous groupings.

Examination of the five HOPACH based clusters revealed minimal overlap with the model based groupings. Cluster 3 corresponded to the Hyp1 up condition with an overlap of 445 probes (41% of the cluster). Similarly, cluster 4, corresponded with the Hyp1 down condition with an overlap of 220 probes (37% of the cluster). However, there were no overlapping clusters with any of the Hyp2 situations. This is not surprising, as hypothesis 2 represents a much more subtle association than Hyp1 and is probably harder to discover.

MA and GO analysis were performed on the five clusters. The MA analysis revealed 8 transcription targets of interests (compared to 19 in the main analysis). Five of these overlapped with those previously identified, suggesting little new information was gleamed. Moreover, it is unclear, based on the cluster results whether the genes are up-regulated or down regulated, making a biological follow-up less clear.

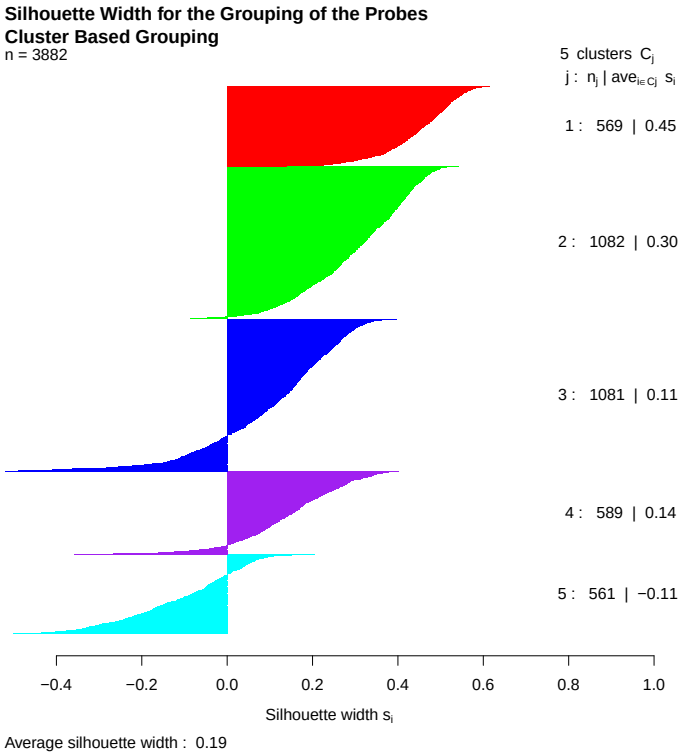


Figure 4.6: Silhouette distances based on HOPACH clustering.

For the GO analysis only clusters II, III & V had significant GO-terms. Cluster III, which shared probes in common with Hyp1, also shared the terms relating to phosphorylation. Clusters V had the most amount of significant terms (15) but most were fairly general (e.g. intracellular part, cell cycle & development process) and did not hint at specific mechanisms. Cluster II had only 3 significant terms, with the most intriguing one being "regulation of nitrogen metabolic process." In total, the GO analysis from the model based analysis was more illuminating.

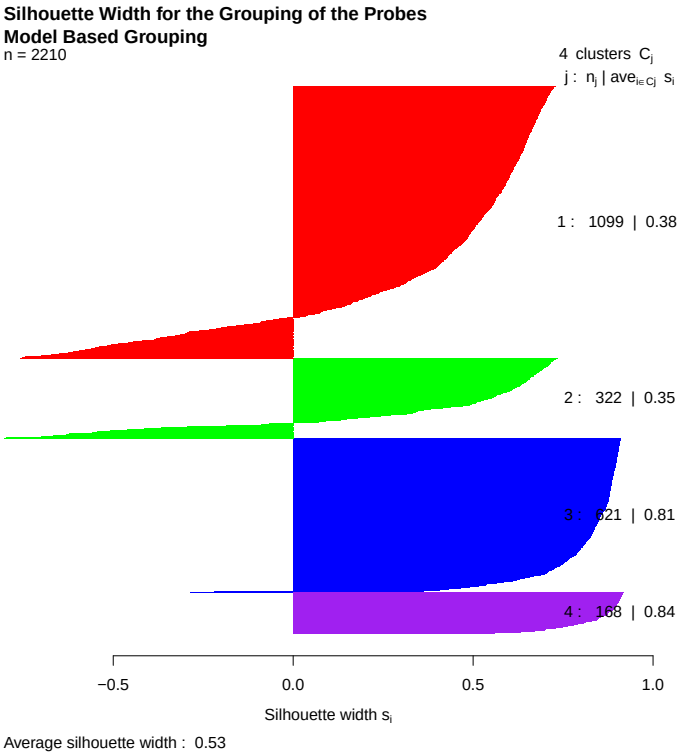


Figure 4.7: Silhouette distances based on model based groupings. Groups 1 & 2 correspond to Hyp1 up and down regulated respectively. Groups 3 & 4 correspond to Hyp2a up and down respectively.

4.4 Discussion

This analysis represents a direct approach towards detecting differentially expressed genes to pass on for secondary analysis. Since every gene expression experiment is different, differing hypothesis will be generated. Models that can best capture those hypotheses will be most successful in detecting gene groupings of interest.

In this analysis, an unexpected pattern in lifespan was detected suggesting an interaction between two genes (Daf-2 & S6k). Four potential hypotheses were derived and an appropriate linear model was fit to the data to test them. Probes were grouped based on whether they conformed to the hypotheses. Such groupings were ultimately more homogeneous than had one used a sophisticated clustering algorithm. While clustering is useful when no clear hypothesis is able to be formulated, this was not the case in this situation.

The use of GO and MA as secondary analyses tools was able to provide new targets for future study and helped explain the underlying biology. While these analyses are fairly qualitative, when used carefully, valuable information can be extracted.

Various TF were identified as potentially up-regulating genes involved in the lifespan extension. Nineteen different genes were tested to see if suppression of these TF could halt the lifespan extension. Of them, 12 (63%) showed an ability to change lifespan extension. Of particular interest are eight genes that significantly change lifespan in the DM.

GO analysis also proved to be informative. Two potential areas of further exploration were identified: phosophoralytion processes and carbohydrate processes. It was also encouraging that genes known to be associated with aging were part of the identified gene sets.

Ultimately the power of this study was the well designed biological experiment. Two separate baselines groups were used, allowing the replication of results and adding a degree of robustness. Moreover, the ability to follow-up findings in a wet lab allowed for the confirmation of the statistical results. Obviously not all studies will be so designed and the specific analysis plan will need to be modified for the particular experiment and question of interest. In the end, this analysis highlights the need to not approach micro-array experiments in a cookie cutter way, but target and change the analysis for the particular study. It is then that meaningful results are obtained.

Bibliography

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57: 289–300, 1995.

D Chen, B. A. Goldstein, A. E. Hubbard, S. Melov, and P. Kapahi. Synergistic lifespan extension by inhibition of both daf-2 and s6 kinase in c. elegans. *Science*, submitted.

M.K. Das and H.K. Dai. A survey of dna motif finding algorithms. *BMC Bioinformatics*, 8, 2007.

M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95:14863–14868, 1998.

S. Grossmann, S. Bauer, P.N. Robinson, and M. Vingron. Improved detection of overrepresentation of gene-ontology annotations with parent child analysis. *Bioinformatics*, 23:3024–3031, 2007.

M. Hollander and D.A. Wolfe. *Nonparametric Statistical Methods*, pages 334–335. John Wiley & Sons inc., New York, second edition, 1999.

C. Linhart, Y. Halperin, and R. Shamir. Transcription factor and microrna motif discovery: The amadeus platform and a compendium of metazoan target sets. *Genome Research*, 18:1180–1189, 2008.

G.A. Mack and J.H. Skilling. A friedman-type rank test for main effects in a two-factor anova. *Communications in Statistics - Simulation and Computation*, 75: 947–951, 1980.

P.J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 1987.

G.K. Smyth. Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Statistical Application in Genetics and Molecular Biology*, 3, 2004.

G.K. Smyth. *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, chapter Limma: linear models for microarray data. Springer, New York, 2005.

M.J. van der Laan and K.S. Pollard. A new algorithm for hybrid hierarchical clustering with visualization and the bootstrap. *Journal of Statistical Planning and Inference*, 117:275–303, 2003.

T. Werner. Cluster analysis and promoter modelling as bioinformatics tools for the identification of target genes from expression array data. *Pharmacogenomics*, 2: 25–36, 2001.

Chapter 5

Conclusion

These studies represent the value of asking specific questions about the data. When different questions are asked about different datasets, different methods will be necessary. Along the way, I also hope to have shown a unity of approach. I have developed an interest (and hope expertise) in the application of statistical learning methods. When used intelligently these methods represent a flexible and non-parametric means of analyzing data. However, when used carelessly the results may be at best muddled or at worst misleading.

The question of determining which genes are related to disease is extremely challenging. I do not suggest that these are the only or even necessarily the best approaches. While the goal of these studies is inference, they are all exploratory in nature. With that in mind it is better to utilize multiple approaches to interrogate the data rather than dogmatically rely on one method. As more hypotheses are derived, we can then progress to a state of performing targeted analyses that allow us to get honest inference and assess causality.

Until we reach that point there is still much important and interesting work to be done in statistical genetics. I eagerly await the new datasets that will find their way to me, as I'm sure they will inspire new questions and new methods.

Appendix A

Some Theory Behind Random Forests

A.1 Introduction

The purpose of this section is to enlighten some of the theory behind the Random Forests (RF) algorithm[1]. RF is in many ways a quintessential black box algorithm with many moving parts which spits out a series of "answers." However, underlying it is a build-up of some simple theory which helps in understanding how best to optimize it. This write-up is written specifically to understand the use of RF for the type of classification problems one would encounter in large genetic associations studies. While most of the discussion is generally applicable, some of the it (particularly that of Section A.2.1) will differ when the outcome is continuous.

A.2 The Components of the Random Forests Algorithm

A.2.1 Bias - Variance Decomposition

One of the first steps in understanding a predictor is to see how its predictions contribute to bias and variance. We start with the setup, given an outcome y, input vector $\mathbf{x}$, relationship $y = f(\mathbf{x}) + \epsilon$, prediction $\hat{f}(\mathbf{x}|T)$ and training set T, the well known decomposition for prediction error (PE) under squared-error loss with a continuous outcome is:

[1]This was initially written in preparation for my qualifying examination (which focused a great deal on RF). I found the theory behind RF quite interesting, particularly the literature on 0-1 loss (section A.2.1) and decided to include it as an Appendix.

$$E_T[y - \hat{f}(\mathbf{x}|T)]^2 = \underbrace{E[\epsilon|\mathbf{x}]^2}_{\text{Noise}} + \overbrace{[f(\mathbf{x}) - E_T\hat{f}(\mathbf{x}|T)]^2}^{\text{Bias}} + \underbrace{E_T[\hat{f}(\mathbf{x}|T) - E_T\hat{f}(\mathbf{x}|T)]^2}_{\text{Variance}} \quad \text{(A.1)}$$

The first term is the variance of the outcome y ($E[\epsilon|x]$ is assumed to be 0) and is referred to as the noise. This represents the irreducible error. The next two terms represent the reducible error. The first of these is the bias. We can think of the bias as the systematic difference between the prediction and the target. The final term is the variance. It is the measure of randomness of the prediction. It is important to note that the variance is independent of the true outcome y and the true function $f(\mathbf{x})$.

In classification with a 0-1 outcome we are trying to minimize $P(\hat{f}(\mathbf{x}) \neq y), y \in \{0, 1\}$ This is usually done under miss-classification loss

$$l(y, \hat{f}(\mathbf{x})) = \begin{cases} 1 & \text{if } y \neq \hat{f}(\mathbf{x}), \\ 0 & \text{if } y = \hat{f}(\mathbf{x}). \end{cases} \quad \text{(A.2)}$$

In the mid 1990s multiple authors attempted to define a decomposition for 0-1 loss [Dietterich and Kong, 1995, Kohavi and Wolpert, 1996, Breiman, 1996a, Tibshirani, 1996]. Most of the effort centered around trying to find a decomposition that was additive in the components of noise, bias, and variance, as in (A.1). Each author proposed a slightly different decomposition, depending on which properties they hoped to satisfy.

Unfortunately a simple thought exercise shows that bias and variance are not additive when the goal is classification. First, note that if a classifier predicts the correct class, $P(\hat{f}(\mathbf{x}|T) = y \geq .5)$, when the true class is class 1, it is unbiased at $\mathbf{x}$. If we have an unbiased classifier, we would desire for the classifier to also have low variance. However, if the classifier is poor, $P(\hat{f}(\mathbf{x}|T) = y < .5)$, we say it is biased. In this scenario we would actually desire the classifier to have high variance because we want to increase the chance that the classification "flips." In this sense, to minimize PE, we see that for an unbiased (good) classifier we want low variance, but for a biased (poor) classifier we want high variance.

Friedman [1997] recognized this interaction between bias and variance. After averaging over all training sets, he decomposes the relationship as,

$$P(\hat{f}(\mathbf{X}) \neq y) = |2f(\mathbf{X}) - 1|P(\hat{f}(\mathbf{X}) \neq f^*(\mathbf{X})) + P(f^*(\mathbf{X}) \neq y) \quad \text{(A.3)}$$

where $f^*(\mathbf{x})$ is the Bayes classifier, and $\mathbf{X}$ designates over all inputs, x. Friedman referred to $P(\hat{f}(\mathbf{X}) \neq f^*(\mathbf{X}))$ as a decision boundary error. Making the simplifying assumption that $P(\hat{f}(\mathbf{X}))$ is normal, he showed this boundary could be represented by:

$$P(\hat{f}(\mathbf{X}) \neq f^*(\mathbf{X})) = \Phi\left[sign(1/2 - f(\mathbf{X}))\frac{E\hat{f}(\mathbf{X}) - 1/2}{\sqrt{var(\hat{f}(\mathbf{X}))}}\right] \tag{A.4}$$

The "boundary bias" is then represented by $sign(1/2 - f(\mathbf{X})(E\hat{f}(\mathbf{X}) - 1/2)$. It is clear that when one predicts to the correct class, "boundary bias" is negative, and PE decreases. Moreover as $[E\hat{f}(\mathbf{X}) - 1/2]$ increases, PE decreases only when one predicts to the correct class. Furthermore, it is evident that decreasing the variance of the predictor is only beneficial when one is on the correct side of the boundary. In this way we see the strong multiplicative interaction between bias and variance.

Gareth [2003] followed this up by suggesting a unified bias-variance decomposition, applicable to all symmetric loss functions. Specifically, he recognized that there is both bias, and the effect due to bias. Similarly there is variance and the effect due to variance. He showed that under squared-error loss these are equal, under other losses they are not.

In genetic studies, we are often not interested in prediction but instead variable importance. It is then natural to ask why concern ourselves with these issues. The concern arises when it comes to tuning our algorithm. In the classification setting, one may not be interested in predicting the class outcome, but instead the underlying probability. Friedman [1997] showed that in the probability estimation setting the bias-variance again becomes additive (assuming squared loss). In an application to K-Nearest Neighbors (K-NN), he demonstrated that the implication of this is that different tuning parameters will be favored depending on the task at hand.

In RF (and all machine learning algorithms) there are tuning parameters that need to be chosen. The parameters are chosen by minimizing the prediction error (via cross-validation, out of bag estimation etc.). The criterion used will be the minimization of the miss-classification error rate (since we do not observe the underlying probabilities). As Friedman, shows this will lead to the choice of different tuning parameters.

A.2.2 CART

Underlying Random Forests is the Classification and Regression Tree (CART) algorithm [Breiman et al., 1984]. Trees are an appealing base learner because they present a simple method to represent complex relationships, particularly those that are present in genetic data. Firstly, the tree structure represents a conditional model making it suited for finding interactions and higher order effects. Furthermore, since trees do not assume linearity in effects, instead performing binary splits, it is ideally suited for discovering *recessive* and *dominant* genetic effects. The main type of effect ill suited for trees are in *additive* effects.

The CART algorithm recursively searches for a split that partitions the data in such a way that minimizes a splitting criterion. This is referred to as a "greedy"

search. After a stopping criterion is met, the final splits partition the predictor space into hyper-rectangles. These regions are referred to as leaves or terminal nodes of the tree.

In the case of classification the splitting criterion, $Q_m(T)$, is typically the gini-index (though other convex losses can be used). For a node m, in region R_m, with N_m observations, we define

$$Q_m(T) \equiv \hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} I(y_i = k)$$

The gini index is then:

$$\begin{aligned} GI &= \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \\ &= 2p(1-p) \text{ when k} = 2 \end{aligned} \tag{A.5}$$

One nice feature of this criterion is that it is prefers pure nodes, unlike miss-classification loss.

Since a fully grown tree, T_0, will have high variance, (changes in the training data will lead to different tree structures), trees are typically pruned, by finding the sub-tree $T_\alpha \subseteq T_0$ which satisfies the criterion:

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha|T| \tag{A.6}$$

where $|T|$ is the number of terminal nodes in the tree and α is a tuning parameter chosen typically by cross-validation. However, in Random Forests, the trees are not pruned but kept at their maximal depth. This results in each tree having low bias, but high variance. This variance is alleviated by bagging (see next section).

A.2.3 Bagging

Brieman proposed *Bagging* (**B**ootstrap **Agg**regat**ing**) as a solution to the instability observed in classifiers such as CART trees [Breiman, 1996b]. In ensemble methods such as bagging, the algorithm used (i.e. CART) is referred to as the "base learner." Bagging is a simple procedure where one selects successive bootstrap samples of the data, (X^B, Y^B) and gets a prediction, $\hat{f}(x^B)$, on each of these samples. The final prediction, $\hat{f}_{bag}(\mathbf{x}|T)$ is determined by either averaging each of the predictions, $\frac{1}{B}\sum_{b=1}^{B} \hat{f}^B(\mathbf{x})$ (for a continuous outcome), or taking a majority vote, $argmax_k \hat{f}_{bag}(\mathbf{x})$ (for classification). To estimate the $P(f(\mathbf{x}) = k)$, the intuitive approach is to average the probability estimate of each of the base learners (in CART this would be the terminal nodes). However the better approach to estimate this quantity is to divide

the number of bagged samples that vote for class k by the total number of bagged samples (see Hastie et al. [2009] pg.286 for discussion).

The motivation behind bagging is to simulate having multiple training sets. If one had access to all training sets, T, then there would be no variance in the final prediction. Bagging then works by reducing the variance of the final predictor. Bühlmann and Yu [2002] and Friedman and Hall [2007] each showed that bagging works via smoothing out first order and higher order variance terms. With respect to bias, since the distribution of $(X^B, Y^B) \sim (X, Y)$, the bias of $\hat{f}_{bag}(\mathbf{x})$ equals the bias of $\hat{f}(\mathbf{x})$, so there is no (asymptotic) increase in bias induced by bagging, though there can be in finite samples.

Another perspective on bagging is that manipulation of the input space is able to increase the search space for an optimal solution [Dietterich, 2000a]. Breiman [1996d] argued that this process works best with unstable predictors. He defined an unstable procedure as one where a small change in the data can lead to large changes in the prediction. He showed that procedures like CART are unstable while procedures such as KNN are stable.

In the case of classification, Breiman [1996b] argued that bagging is effective in the case of order-correct predictor which he defined as

$$argmax_k f(k|\mathbf{x}) = argmax_k P(j|\mathbf{x}) \tag{A.7}$$

This simply means that if a classifier predicts class k for a given input $\mathbf{x}$ then the aggregated classifier will also predict class k, i.e. is unbiased at $\mathbf{x}$. Similar to the previous discussion of the bias-variance relationship for classification, Breiman noted that for good classifiers bagging can be very useful, but for bad ones, bagging can actually be harmful. This is because for good classifiers we want to decrease the variance (to reduce overall PE) but for bad classifiers we want to increase the variance (to reduce the overall PE).

Bagging Type

Bühlmann and Yu [2002] and Friedman and Hall [2007] also both showed that $m < n$ sampling without replacement, where $m = n/2$ is just as effective as bagging with replacement, and computationally more efficient. Bühlmann and Yu referred to this as *subagging* (**sub**sample **agg**regat**ing**).

Dietterich [2000a] notes that large datasets don't see the same benefits from bagging as do smaller ones because each bootstrap sample is more similar to each other than with a smaller data set. Subagging would serve three benefits. First would be computational - since fewer observations are used in growing the trees. The second is through a reduction in tree correlation - the trees would be more different from each other. The third is in a reduction of tree size. This third component decreases the degrees of freedom of the final model.

Out-Of-Bag Error-rate

One of the appeals of bagging is that it presents a computationally efficient means to calculate the generalized error (GE), the PE of $\hat{f}(\mathbf{x}|T)$ on training set T'. The best way to calculate GE is on an independent validation set. In lieu of one, the most typical approach is to use V-fold cross-validation (CV).

For bagged learners, CV is computationally very expensive. However in each iteration of bagging, approximately 37% of the sample is not part of the bootstrap sample.

$$\begin{aligned} P(\text{observation } i \in \text{bootstrap sample } b) &= 1 - (1 - \frac{1}{N})^N \\ &\approx 1 - e^{-1} \\ &= 0.632 \end{aligned}$$

Breiman [1996c] showed that this Out-Of-Bag (OOB) sample can be used as a test set to get a measure of error. Over the entire bagging run, this error can be aggregated for each input vector $\mathbf{x}$. Some authors have referred to this as monte-carlo cross-validation (e.g. Dudoit and van der Laan [2005].) Wolpert and Macready [1999] showed that this can provide a more stable estimate of GE than typical V-fold CV. If we define C^{-1} as the set of indices not in bootstrap sample b, and $|C^{-1}|$ as the number of such samples, the OOB estimate becomes:

$$\widehat{GE}_{OOB} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|C^{-1}|} \sum_{b \in C^{-1}} L(Y_i, \hat{f}^B(x_i)) \tag{A.8}$$

Since each bootstrap sample will have a sample size of about $.632N$, this estimate will behave similar to 2-fold CV.

Of particular interest, is that the OOB error-rate provides a convenient means to choose tuning parameters in RF. As will be discussed, RF involves the choose of multiple tuning parameters, and one can simply choose the settings that minimize the OOB error-rate.

A.2.4 Randomization

A final method for improving ensemble learners is by injecting randomization into the base learner. Many different procedures have been explored for this [Dietterich, 2000a]. Like bagging, the author showed that this randomization is able to expand the search space and alleviate what they termed the "statistical" burden. In another study, he showed that injecting randomization can be more effective than bagging for large datasets [Dietterich, 2000b].

As noted in Hastie et al. [2009] the variance reduction induced by bagging is

limited by the correlation between the trees, since the trees are not independent, only identically distributed. If we denote the variance of each tree as σ^2 and the correlation between trees as ρ the variance of the average is:

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2 \tag{A.9}$$

As the number of bootstrap iterations increases, the second term goes to 0, and we are left with the correlation between trees, ultimately limiting the benefits of the bagging process. As dataset size increases, the correlation between bagged samples increases, decreasing the effect of bagging. Injecting randomization into the tree growing process serves to further de-correlate the trees, further reducing the variance. Different proposals have been made for injecting this randomization and the Random Forests algorithm is actually only one example of randomized tree algorithms (see Cutler [1999]).

A.2.5 The Random Forests Algorithm

At this point we can consider the RF algorithm, proposed by Leo Brieman in 2001 [Breiman, 2001]. At its core, it is bagged CART trees, with injected randomization.

The first alteration is that instead of pruning the CART trees, they are grown to maximal depth. These fully grown trees will be fairly unbiased but will be highly variable (recall we prune trees in CART to eliminate the variance). This variance is reduced via the randomization.

In RF, the randomization comes in the tree growing process. Before each split, one chooses a subset $m \leq p$ of the number of predictor variables to search over. The choice of m, denoted *mtry*, is the primary tuning parameter. The smaller *mtry* the less correlation between trees and the greater the potential variance reduction via bagging is possible. However, smaller *mtry* will also lead to more biased trees, hence resulting again in the classic bias-variance trade-off.

RF reduces PE only through a reduction of variance, as the bias stays the same (or gets a bit a worse). Breiman shows that unlike other methods (notably Boosting), RF does not over fit as the number of trees increases. However, as noted in Hastie et al. [2009] in the limit it can overfit.

A.3 Variable Importance

When applying RF for classification there are two primary forms of Variable Importance: permutation importance & gini importance.

A.3.1 Permutation Importance

The permutation importance (pVI) is the increase in misclassification for Out-Of-Bag (OOB) person i after variable j has been permuted. If we consider the quantities:

- n_{ijk} = number of trees that split on variable j and misclassify observation i
- m_{ijk} = number of trees that do not split on variable j and misclassify observation i
- pn_{ijk} = number of trees that split on variable j and misclassify observation i when variable j is permuted
- pm_{ijk} = number of trees that do not split on variable j and misclassify observation i when variable j is permute

We can represent it as:

$$\begin{aligned} pVI_{ijk} &= (pn_{ij} + pm_{ij}) - (pn_{ijk} + pm_{ijk}) \\ &= pn_{ijk} - n_{ijk} \text{ since } pm_{ij} = m_{ij} \end{aligned}$$

and we can calculate:

$$\begin{aligned} pVI_{ij} &= \frac{1}{ntree} \sum_{k=0}^{ntree} pn_{ijk} - n_{ijk} \\ pVI_{jk} &= \frac{1}{np} \sum_{i=0}^{np} pn_{ijk} - n_{ijk} \\ pVI_{j} &= \frac{1}{np \times ntree} \sum_{i=0}^{np} \sum_{k=0}^{ntree} pn_{ijk} - n_{ijk} \end{aligned} \tag{A.10}$$

The three quantities in (A.10) represent respectively the importance of variable j for person i, the importance of variable j for tree k and the overall importance of variable j. Each representation will have different utility depending on the question of interest.

pVI has some nice properties. Since it is calculated off of the OOB sample, it can be viewed as the predictive quality of that variable. A variable with no importance would be expected to have $E(pVI) = 0$ since permutation should neither increase nor decrease misclassification. There is also a notion of a population level effect of the variable importance since the probability of being permuted to a different value is determined by the observed population.

Adele Cutler (Personal Communication) proposed a T-like statistic where one calculated the SE(pVI) across people[2]. In practice this measure is not always successful,

[2]Since people are assumed independent and trees are only conditionally independent it is more appropriate to calculate the standard error using pVI_{ij} rather than pVI_{jk}

and she is uncertain of its utility. It is also applicable for any outcome or predictor type.

Correcting Permutation Importance

An important consideration with applying RF to GWA data is the large degree of correlation (referred to as LD) among SNPs. There are a couple of ways to formulate the problem in calculating VI induced by correlation. Being a greedy algorithm, RF searches over all variables. In calculations of VI, this creates a smoothing and shrinkage of all VI measures - in an analogous way to Ridge regression. This creates problems for correlated variables as the relative importance is diminished. Another formulation is that since VI is calculated from the number of trees for which a variable appears two SNPs that are in perfect LD will appear in trees about half as often as each individual one may appear by itself, effectively lowering the VI of each SNP. While this does not present a problem for prediction, it can skew the VI rankings.

Genuer et al. [2008] examined the impact of correlated variables, and found that as the number of variables correlated with a true causal one increased, the variability of the true causal one increased and its average importance decreased. Similar effects were noted by other authors, notably Strobl et al. [2007].

Two proposals have been made for correcting for this analytically. pVI is calculated by dividing pVI_{ij} by the total number of trees in the forest. Meng et al. [2009] suggested dividing by the total number of trees of which variable j is a member. This has the appeal that two perfectly correlated variables will no longer "take away" from each other. In practice, with large p this leads to highly unstable pVI measures. This works best with less sparse solutions or smaller p where all variables have a chance to be brought into the model.

Strobl et al. [2008] suggested using a conditional permutation scheme to calculate VI. One empirically determines which variables are correlated with the variable of interest, and uses the partitions in the individual tree to permute the variable of interest within blocks. While effective when p is small, this has drawback of creating a VI measure that is more computational and not uniform across trees.

A.3.2 Gini Importance

The gini importance (gVI) is the second primary form of RF VI. Unlike pVI, gVI is only applicable in the case of classification. The gini index (GI) is the criterion used when growing the trees in RF for classification. Recalling (A.5), for binary classification,

$$GI = 2p(1-p)$$

where p is the proportion in the second class. The split which minimizes GI is the preferred split. If we index the node for a given tree by n, we can then define:

$$
\begin{aligned}
gVI_{jkn} &= (GI_{parent} - GI_{\text{daughter left}} + GI_{\text{daughter right}})np_{kn} \\
gVI_{jk} &= \sum_{n_j \in Tree_k}^{N} gVI_{jkn} \\
&\text{(summing over the nodes containing variable } j \text{ in tree } k) \\
gVI_j &= \frac{1}{ntree} \sum_{k=1}^{ntree} gVI_{jk}
\end{aligned}
\tag{A.11}
$$

gVI_{jk} directly measures the importance of variable j to tree k. The higher the value the better the variable was in splitting the data. In this sense it is very different from pVI. There is no notion of out of sample testing. Instead gVI_{jkn} can be thought of as a χ^2 test, conditional on what has already occurred in the tree (for the root node it is conditional on nothing).

Another property is that $gVI_j \geq 0$ with equality if variable j does not appear in any tree. Like pVI it will have trouble with correlated variables but can also be corrected by weighting. Since gVI is calculated based on the in-sample data it does not have a population level interpretation as pVI. Instead gVI only considers the relationship between the variable and the model.

pVI is the more commonly used form of VI. However, some intuition shows that gVI can be a preferential VI measure when the predictive quality of the predictors is low (i.e. OOB-ER $\approx$ 50%). Since pVI is calculated based on the increase of misclassification after permuting variable j, if the baseline misclassification rate is already relatively high, there is little chance for permutation to make prediction worse. This will lead to a uniformly low pVI. Conversely, since gVI is calculated relative to the grown tree it does not suffer this problem. It is easy to show this via simulation. However, since variables have to be in the tree, there will always be variables with high gVI and it is questionable how "important" a variable is that doesn't improve prediction. Moreover, it is much more challenging to consider distributional properties for gVI.

A.3.3 Determining Important Variables

Once a VI measure has been decided upon, the next challenge is determining how many variable are actually important. Ideally, one would be able to determine statistical properties for the VI measures to determine when the observed value differed from an expected value. Some work has been performed in this area (e.g. Adele Cutler's T-test) but to this point no formal approach has been adopted.

In SNP studies we are often trying to determine which variables are worthy of future follow-up. Generally we only look at the rank order of the top variables and

decide to follow-up on those. This has led to a range of ad-hoc procedures. Díaz-Uriarte and Alvarez de Andrés [2006] suggesting removing the bottom 10% and re-running until prediction decreased. In Goldstein et al. [2010] a procedure motivated from principle components analysis was used. The scree plots of the VI measures were examined and the cutoff occurred at the "elbow," typically 25 SNPs. However this is an unsatisfactory solution and ideally some objective cut-off could be determined.

A.4 Applying Random Forests

A.4.1 Tuning Parameters

Running RF involves the choice of two primary tuning parameters: *mtry* and the number of trees (*ntree*). One of the appeals of RF is that it is considered relatively robust to tuning parameter settings, make it an effective "off-the-shelf" algorithm.

Using the OOB Error-Rate

The OOB error-rate provides an unbiased estimate of the generalized error. Minimizing this error allows one to select the optimal tuning parameters to generate the best predictive model. However, when one begins augmenting the dataset (e.g. removing unimportant variables) the OOB error-rate is no longer an unbiased estimate of the generalized error, though its minimization can still be used for tuning parameter selection [Svetnik et al., 2004].

There is less of a guarantee that minimizing the OOB error-rate will provide the optimal results when one's goal is VI. However, theoretical work has shown that the prediction error can be tied directly to the strength of association of the set of predictors (see Chapter 3). Furthermore, my own internal testing has shown that as the OOB-ER improves, the quality of the VI rankings improve. With this in mind, one should interpret VI in conjunction with the OOB error-rate. If the OOB error-rate, is close to the null value (50%) it is likely that none of the predictors are associated with the outcome, regardless of the VI scores.

mtry

mtry is the primary tuning parameter and the default recommended value of is $\sqrt{p}$ where p is the number of predictors. However, in genetic studies, we expect that the vast majority of the input variables are simply noise. Therefore, if *mtry* is too small the chance of selecting an important variable to search over at a given node will be small.

Díaz-Uriarte and Alvarez de Andrés [2006] examined *mtry* values ranging from $\sqrt{p}$ to $13\sqrt{p}$ (it was unclear why they didn't test *mtry* = p) and generally found that the larger the *mtry* the better. Genuer et al. [2008] noted that *mtry* is more important

for VI calculation than for prediction and that with sparse data, *mtry*= p leads to greatest stability. In Chapter 2, working with a large SNP dataset ($p > 330,000$) found that *mtry* values much larger than $\sqrt{p}$ were needed. In simulation work, the OOB error rate and VI measures were fairly similar to *mtry* values of $.1p$ and p, indicating that the setting is fairly robust to a sensible choice.

mtry has its greatest impact on the final sparsity of the solution, as larger values of *mtry* lead to fewer variables brought into the tree. Moreover, the smaller the *mtry* the larger the individual trees. In this way it can be thought of as controlling the degrees of freedom (df) of the model, with the higher *mtry* the fewer df used. This actually presents a slight paradox. As discussed, the purpose of decreasing *mtry* is to lower the correlation between trees, and hence allow bagging to reduce variance. Therefore, we recognize another complicating factor in the variance of RF: reducing *mtry* can both decrease the variance via de-correlation and increase the variance via larger trees and increased model complexity.

For genetic data, where we expect most of the predictors to be noise, this suggests that we want to find ways to reduce the tree size. While increasing *mtry* is one means to do so, this will not be the most effective means and other ways should be sought (see below).

Number or Trees

Another important consideration is how many trees to grow. This appears to be a data set dependent factor, where stronger predictors lead to quicker convergence. Glaser et al. [2007] using a much smaller data set (20 SNPs), grew forests with up to 5000 trees, and found that after 400 trees, the results were stable. Díaz-Uriarte and Alvarez de Andrés examined forest sizes ranging from 1000 to 40,000 and found no differences in error-rates. While for prediction purposes, few trees are necessary and the OOB error rate will generally converge rapidly, Genuer et al., noted that for variable importance more trees will generally lead to refinement and stability in variable importance.

If the only concern is prediction, one can simply use the OOB error-rate as guide to determine *ntree*. For VI this is not the case. Both my simulation and empirical work suggested that growing forests much larger ($2\times - 3\times$) the point of stable OOB error-rate was necessary.

Terminal Node Size

Terminal node size (*nsplit*) is not an often discussed tuning parameter. The setting of *nsplit* tells the algorithm when it should stop splitting the node leaves of the tree. In CART one grows trees to maximal depth and then prunes nodes using cross-validation. As mentioned in RF this pruning step is skipped. There are both computational and theoretical reasons for this. Computationally, pruning is expensive

and potentially impractical over a forest of trees. Theoretically, though, pruning is unnecessary. From a bias-variance perspective, the purpose of pruning is to increase the stability (i.e. lower the variance) at the cost of increased bias. The increased variance of unpruned trees is due to their tendency to overfit the data. However, the bagging process, specifically aims to reduce variance (with no cost to bias) and avoid overfitting making pruning superfluous. In addition Breiman [1996d] showed that bagging works best with weak base learners, of which unpruned trees are one.

In practice, *nsplit* is a tuning parameter that has probably not received enough attention. The primary evaluation of it was by Segal [2004]. As with *mtry* the conclusion was that there was often an optimal *nsplit* though growing trees to maximal depth did not lead to overfitting. One area, in which, it may be of greater importance to consider *nsplit* is in the calculation of proximities (see Section A.4.3).

A.4.2 Modifying the Data

One can also effect the final solution by modifying the input the data.It is important to note, that once the data is modified the OOB-ER no longer represents an unbiased estimate of GE [Svetnik et al., 2004].

Correlation

There are two approaches for dealing with this issue: computationally and in the set-up of the data. Computational approaches are discussed in Section A.3.1. A second approach, applicable with genetic data, is to pre-process the data based on LD structure. This was the approach taken in Goldstein et al. [2010]. It was empirically found that pruning to an LD level of 90% resulted in identifying different interesting variables and did not seem to degrade PE.

Removing Unimportant Variables and Important Ones

As mentioned, the sparsity of the final model is a function of both *mtry* and *ntree*. It is desirable to remove these sparse results as they likely represent noise and will simply make finding the optimal solution more challenging. Díaz-Uriarte and Alvarez de Andrés outlined a strategy of sequentially removing genes by dropping the bottom 20% or 50% performing successive runs until there was a noticeable increase in PE. In Chapter 2, I removed the SNPs with VI = 0. This obviously is not a universal strategy as not all applications will have VI values of 0. As with Ridge regression, most variables will have a chance to have at least some importance. Therefore RF is likely not the best algorithm for removing unimportant variables. For such scenarios, algorithms that provide a sparser solution should be utilized.

In the application to the MS data I simply removed the SNPs with 0 VI. This obviously is not a universal strategy as not all applications will have VI values of 0.

As with Ridge regression, most variables will have a chance to have at least some importance. Therefore RF is likely not the best algorithm for removing unimportant variables. For such scenarios, algorithms that provide a sparser solution should be utilized.

Not only can we consider removing unimportant variables, we can also consider removing overly strong results. Since RF searches over multiple markers looking at joint and conditional effects, a strong marker or set of strong markers, could over shadow weaker yet important effects. The work with the MS showed that after removing Chr 6p, new variables were found that would not have been found otherwise. This is an important consideration that has not been discussed elsewhere.

A.4.3 Other Uses of Random Forests

The derived collection of trees provides a significant amount of information about the complex relationships between predictors and observations. This information can be exploited for many additional uses including clustering, imputing missing data, detecting which observations are related (*proximities*), detecting outliers and graphing. Most of these are detailed on the main website for Random Forests [Breiman and Cutler, 2010]. Cutler and Stevens [2006] provide an overview of some of these uses for genetics applications.

While many of these methods are implemented in most versions of RF, few of these have seen application. Some of these methodologies (e.g. imputing and outlier detection) are probably better accomplished via other methods that take better advantage of genetic structure. However, other analyses (e.g. clustering and proximities) do have the potential to provide insight into the structure of genetic data. However, the primary aspect limiting their use is the relative weakness of genetic data. Many of these analyses exploit very subtle relationships in the tree structure. Since genetic data is often weakly predictive [Clayton, 2009] there is often not enough information content to accurately define these relationships. However these are methods worthy of further exploration.

A.4.4 Implementations of Random Forests

To date there are a number of different implementations of RF. Since there is always potential for implemented algorithms to get lost in translation we make no statement about each's accuracy as we have not used all of them. However, each has slightly different features and may be suitable for different data problems. The original code was written in Fortran by Leo Breiman and Adele Cutler and is available on their website [Breiman and Cutler, 2010]. Their code was adapted for use within the R environment by Andy Liaw and Matthew Wiener in the package randomForest. Other R packages have been created as either add-ons (e.g. varSelRF) or as amendments of the RF algorithm (e.g. cforest). The original Fortan code was licensed to Salford

Systems, and implemented with a GUI and is available for a licensing fee. Numerous open source versions are available, most geared towards handling large data problems. Possibly, the most developed is Random Jungle [Schwarz et al., 2010] implemented in C++ and able to handle GWA data.

A.5 Other Classifiers

One can get insight into an algorithm's function and utility by comparing it to other algorithms. We briefly mention three here: K-Nearest Neighbors (K-NN), Penalized Regression and Boosting.

A.5.1 K-Nearest Neighbors

K-Nearest Neighbors (K-NN) goes back to the 1950s and is one of the simplest classification algorithms to implement. For simplicity, assume the input vector $\mathbf{x_0}$ is real valued, one calculates

$$d_i = \|x_i - x_0\| \tag{A.12}$$

The K is a tuning parameter that determines how many neighbors to consider, ordered by d_i. The classification for x_0 is the majority vote of those K. A primary limitation of K-NN is that it provides little insight into VI.

Lin and Jeon [2006] compared RF to an adaptive form of K-NN. Particularly for classification, the relationship to K-NN stems from the fact that trees are grown to maximal depth, where often there will be only one class in the terminal node of a tree. Therefore over the forest of trees, the classification for a new observation will be a weighted version of a certain number of neighbors.

A.5.2 Penalized Regression

Penalized regression [Hastie et al., 2009] is an important class of algorithms that has increased in popularity with improved computational "tricks." For classification this is done in the context of logistic regression. The general equation to optimize is:

$$max_{\beta_0,\beta}\Big\{\sum_{i=1}^{N}[y_i(\beta_0+\beta^T x_i)-\log(1+e^{\beta_0+\beta^T x_i})]-\lambda\sum_{j=1}^{p}|B_j|^{\alpha}\Big\} \tag{A.13}$$

α is a user set value that controls the type of penalty, while λ is a tuning parameter to optimize the function. When $\alpha = 1$ the penalty represents an ℓ_1 penalty and the algorithm is referred to as the LASSO. When $\alpha = 2$ the penalty represents an

ℓ_2 penalty and the algorithm is the traditional Ridge regression. The Elastic Net algorithm allows one to vary α between 1 & 2.

λ serves as a tuning parameter that controls the complexity of the model. The appeal of LASSO over Ridge regression is that the LASSO penalty will result in a more sparse solution. While the optimal Ridge solution, like RF, will shrink all coefficients towards 0 with few equal to 0, the optimal LASSO fit will have many coefficients at 0. This makes the LASSO ideal for variable selection.

The LASSO has been successfully implemented with GWA data [Wu et al., 2009]. The one concern with the LASSO, as opposed to RF, is that the method requires specifying a parametric linear model. The application by Wu et al. allows for a search for interaction terms, but it is unclear how successful it is with detecting complex genetic effects. In this sense tree structures are preferable.

The relationship between RF and Ridge regression stems from variable importance. Unlike algorithms such as LASSO and Boosting, which tend to produce a sparse solution, placing weight on only few variables, both RF and Ridge result in shrunken VI measures, allowing many variables to "speak." This is desirable when most variables are associated with the outcome resulting in a more stable solution with emphasis spread across the variables. However, when association is due only to correlation (LD) with a true causal variant, this results in what have been called biased importance scores [Strobl et al., 2007].

A.5.3 Boosting

Boosting has seen a large application in machine learning fields but has had no known applications to genetic data (a pubmed search of the terms "boosting" and "SNP" yielded no results). While there is extensive literature on boosting I briefly mention their appeal as a learner and some thoughts as to why it may not be appropriate for genetic data.

Boosting is an ensemble algorithm that like RF has trees as the base learner. However, unlike RF, these trees are not fully grown trees, often containing only a few nodes (the number of nodes is a tuning parameter). While the trees in RF (and bagging) are identically distributed, the trees in boosting are not. Instead each observation in the training set receives a weight that is updated based on some classification error (generally an exponential loss), with greater error, resulting in greater weight. Therefore each iteration of Boosting, attempts to fit the classifier on those observations which is hardest to classify. By doing this, Boosting is able to both decrease variance (because of the aggregation of classifiers) and bias (by doing a better job on those that are miss-classified).

Multiple studies have shown Boosting to be as good as and often better than bagging and RF [Breiman, 1996a, Dietterich, 2000b, Hastie et al., 2009]. Moreover, Boosting, has similarities with LASSO, in that it tends to result in a much sparser solution than does RF.

However there are two problems with Boosting, one computational and one more systemic. Due to this bias reduction mechanism, Boosting is prone to over-fitting. Therefore there needs to be constant CV to determine when to stop growing trees. Computationally this is very expensive. However, a more systemic problem is that the performance of Boosting degrades quickly with noisy data, particularly compared to bagging and randomization procedures [Dietterich, 2000b]. Therefore while a very attractive algorithm, Boosting is not entirely appropriate for genetic data which contains large samples and many irrelevant variables.

A.6 Conclusion

This appendix walked through the theoretical background of RF, while highlighting some relevant research. While in many ways a black box algorithm, in can also easily be broken down into its components: classification, trees, bagging & randomization. Understanding how an algorithm works, particularly the components that control the bias and variance, allows the user to better control the output via the different tuning parameters.

Having worked with RF, I agree with Brieman that it is a great "off-the-shelf" algorithm. While working with genetic data presents particular challenges, using default settings will generate somewhat reasonable results (though ideal settings exist). The underlying algorithm is relatively fast making it capable of handling large Genome Wide Association studies. The tree structure is well suited for genetic data since it is non-parametric and allows for the existence of conditional and higher order effects.

The simplicity of the RF algorithm makes it well suited for advanced users to manipulate it to suit their own data needs. The two VI measures discussed are only the standard and most general VI forms, but it is easy to conceive of specific VI measures for particular data problems. For example, it would be relatively straight forward to construct an analogous pVI looking at two variables. Moreover, this exposition barely touches on the many subtle questions one can ask after fitting RF. Within the collection of trees there is a lot of information about the relationship between the variables and observations allowing one to explore clustering, proximities and visualization. Experience has shown that these more subtle relationships can only be gleamed when the overall predictor is fairly strong. we That being said, I also have to agree with critics of RF that it is not a perfect algorithm. The VI measure that it produces are inherently ad-hoc and lacks any statistical properties (some work has been attempted to gleam some statistical properties but so far none have been determined). The lack of sparsity makes it ill suited for variable selection. Also, there is a tendency, particularly when the data is not very predictive for the predictions generated from RF to be highly shrunk to the mean.

As the "No Free Lunch" states, there is no perfect algorithm [Wolpert, 1996]. As

with all modeling situations it is important to find the right tool for the job. For large genetic data, RF can be the right tool, though it is does need to be used appropriately and with insight.

Bibliography

L. Breiman. Bias, variance, and arcing classifiers. Technical report, UC Berkeley, 1996a.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996b.

L. Breiman. Out-of-bag estimation. Technical report, UC Berkeley, 1996c.

L. Breiman. Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24:2350–2383, 1996d.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman and A. Cutler. Random forests. `http://www.stat.berkeley.edu/~breiman/RandomForests/`, December 2010.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, 1984.

Peter Bühlmann and Bin Yu. Analyzing bagging. *Annals of Statistics*, 30:927–961, 2002.

D.G. Clayton. Prediction and interaction in complex diseases. *Plos Genetics*, 5, 2009.

A. Cutler. Fast classification using perfect random trees. Technical report, Utah State University, 1999.

A. Cutler and J.R. Stevens. Random forests for microarrays. *Methods in enzymology*, 411:422–32, 2006.

R. Díaz-Uriarte and S. Alvarez de Andrés. Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, 7:3, 2006.

T. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000a.

T. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40:139–157, 2000b.

T. Dietterich and E. B. Kong. Machine learning bias, statistical bias, and statistical variance of decision tree algorithms. Technical report, Oregon State University, 1995.

S. Dudoit and M. J. van der Laan. Asymptotics of cross-validated risk estimation in estimator selection and performance assessment. *Statistical Methodology*, 2:131–154, 2005.

J. H. Friedman. On bias, variance, 0/1—loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1:55–77, 1997.

J. H. Friedman and P. Hall. On bagging and nonlinear estimation. *Journal of Statistical Planning and inference*, 137:669–683, 2007.

J. M. Gareth. Variance and bias for general loss functions. *Machine Learning*, 51(2): 115–135, 2003.

R. Genuer, J.M. Poggi, and C. Tuleau. Random Forests: some methodological insights. Technical report, INRIA, 2008. URL `http://hal.inria.fr/inria-00340725/en/`.

B. Glaser, I. Nikolov, D. Chubb, M. L. Hamshere, R. Segurado, V. Moskvina, and P. Holmans. Analyses of single marker and pairwise effects of candidate loci for rheumatoid arthritis using logistic regression and random forests. *BMC Proceedings*, 1 Suppl 1:S54, 2007.

B.A. Goldstein, A.E. Hubbard, A. Cutler, and L.F. Barcellos. An application of random forests to a genome-wide association dataset: Methodological considerations & new findings. *BMC Genetics*, 11:49, 2010.

T. Hastie, R. Tibshirani, and J. Friedman. *Elements of Statistical Learning*. Springer, New York, 2 edition, 2009.

R. Kohavi and D. H. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *Machine Learning: Proceedings of the Thirteenth International Conference*, 1996.

Y. Lin and Y. Jeon. Random forests and adaptive nearest neighbors. *Journal of the American Statistical Association*, 101, 2006.

Y. A. Meng, Y. Yu, L. A. Cupples, L. A. Farrer, and K. L. Lunetta. Performance of random forest when SNPs are in linkage disequilibrium. *BMC Bioinformatics*, 10: 78, 2009.

D.F. Schwarz, I.R. Knig, and A. Ziegler. On safari to random jungle: a fast implementation of random forests for high-dimensional data. *Bioinformatics*, 26:1752–8, 2010.

M. R. Segal. Machine learning benchmarks and random forests regression. Technical report, CBMB Working Paper, April 2004.

C. Strobl, A. L. Boulesteix, A. Zeileis, and T. Hothorn. Bias in random forest variable importance measures: illustrations, sources and a solution. *BMC Bioinformatics*, 8:25, 2007.

C. Strobl, A.L. Boulesteix, T. Kneib, T. Augustin, and A. Zeileis. Conditional variable importance for random forests. *BMC bioinformatics*, 9:307, 2008.

V. Svetnik, A. Liaw, and C. Tong. Variable selection in random forest with application to quantitative structureactivity relationship. In N. Intrator and F. Masulli, editors, *Proceedings of the 7th Course on Ensemble Methods for Learning Machines*. Springer-Verlag, 2004.

R. Tibshirani. Bias, variance and prediction error for classification rules. Technical report, University of Toronto, 1996.

D. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Computation*, pages 1341–1390, 1996.

D. H. Wolpert and W. G. Macready. An efficient method to estimate bagging's generalization error. *Machine Learning*, 35:41–55, 1999.

T.T. Wu, Y.F. Chen, T. Hastie, E. Sobel, and K. Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–21, 2009.