MACHINE LEARNING FOR BIOSTATISTICIANS: A HYPOTHESIS DRIVEN APPROACH

By

RICHARD T GUY

A Thesis Submitted to the Graduate Faculty of

WAKE FOREST UNIVERSITY

in Partial Fulfillment of the Requirements

for the Degree of

MASTER OF SCIENCE

in the Department of Computer Science

May, 2010

Winston-Salem, North Carolina

Approved By:

Carl D. Langefeld, Ph.D., Advisor _____

Examining Committee:

David John, Ph.D., Chairperson _____

Peter Santago, Ph.D. _____

William Turkett, Ph.D. _____

Table of Contents

List of Tables

List of Figures

Acknowledgments

First, thanks to Elizabeth Glenn for making me a better person and a better scientist. You are my best friend, and Toronto will be an adventure I can't wait to have with you. I look forward to sharing in both of our successes in the years to come.

There are several people without whom I would not be in graduate school, and others without whom I would not be in Computer Science. My parents, Merry and Joseph set the example of academic work; I am merely following in their footsteps. Dr. Dee Parks, Dr. Alice McRae, and Prof. Frank Berry gave me an early start in Computer Science. In addition, James Glenn reintroduced me to the field and is one of the major reasons I am pursuing it farther.

One of the most important influences on my academic work is Dr. Carl Langefeld, and I am deeply indebted for two years of great responsibility and opportunity in his lab. In addition, thanks to the many people with whom I have worked in his group. Joshua Grab, Mary Comeau, Mindy Marion, Lingyi Lu, Julie Ziegler, Paula Ramos, and Kenneth Wilson have all been very helpful and patient.

Pete Santago played a large role in shaping the ideas in this thesis and really deserves to be listed as a co-adviser. David John has proven invaluable in many ways, not just my thesis. In addition, William Turkett helped hone many of the arguments in this thesis, especially from a machine learning perspective.

Three professors have been mentors in some capacity over the last three years. Dr. Todd Torgersen has been an example of the kind of professor I want to be, as well as a great source of inspiration about life in general. Dr. Robert Plemmons helped immensely as I made the transition to Ph.D. Dr. Paúl Pauca has been a mentor, and I look forward to continuing to work with him over the summer and beyond.

Abstract

Richard T. Guy

Advances in microchip technology have enabled genome-wide association studies that attempt to find regions of variation that associate with disease in large scale case/control studies. Traditional statistical methods have demonstrated success, but they have failed to make the transition to identifying large sets of interacting variations. Among the reasons are computational complexity and the curse of dimensionality as complicated regression models are deployed.

Recently, a great amount of attention has been paid to machine learning methods for detecting interacting sets of single nucleotide polymorphisms (SNPs) that associate with genetic disorders. Many of the algorithms that have been applied suffer from two problems. First, each algorithm has been developed and presented on data sets on which it is designed to perform well. Second, they require exhaustive search on high order sets of SNPs and are computationally infeasible for modern sample sizes. Comprehensive studies of algorithm effectiveness have failed to further investigate the claims of the algorithm designers, particularly with respect to type I error. In this thesis, we present a study of several algorithms for the detection of interacting sets of SNPs that have been recently published, including Multi-factor Dimensionality Reduction, Support Vector Machines and Random Forests. We compare their power and type I error to logistic regression, which is a popular statistical method for single SNP tests or small interactions. All three machine learning methods are demonstrated to have elevated type I error of detecting an interaction between two SNPs. In addition, power is shown to differ from logistic regression only when a penetrance model exists that does not fit the regression model used.

We also investigate the use of Alternating Decision Trees and Bagging for the detection of interacting and associated SNPs. We introduce a novel interpretation of bagged ADTrees that allows for detection of sets of SNPs that associate with disease with high positive predicive value. Simulated testing shows that ADTrees have power comparable to other machine learning algorithms with less elevated type I error of detecting an interaction provided that a marginal effect exists in at least one SNP involved in an interaction. Another advantage of ADTrees and ADTrees using Bagging is complexity that is linear in the number of SNPs in the sample, unlike all other methods considered for the detection of pairs of interacting SNPs.

Last, we present a software tool called SNPdoc that was developed to assist in genome-wide association studies by enabling fast aggregation of statistical and genomic information. This will be published as presented in this thesis.

Chapter 1: Introduction

The past decades have seen exponential growth in the ability to sequence and analyze genetic data. The first genome to sequenced, the virus Phi X 174 in 1978 by Fred Sanger, contained 5368 bp (base pairs) in its entire genome. For his work, Dr. Sanger was awarded two Nobel Prizes. With the rise of more powerful computers and technological breakthroughs, the genome of a bacterium called *Haemophilus influenzae* became the first genome of a free living organism to be sequenced in 1995. It contained $1,830$Kb, or barely 0.5% of the genetic material of *Homo sapiens* [9]. Six years later, the first draft of the human genome was released, and subsequent years have seen further advances in genotyping ability. Attention has turned to mapping differences between organisms of the same species. Cost and efficiency improvements have enabled large-scale genetic studies of association with disease. Current state of the art technology from Affymetrix contains 1.8 million probes and can record genetic information at over $900,000$ locations at a cost of a few hundred dollars.

One of the key advances made possible by large-scale human genotyping is the ability to pinpoint the causes of many genetic diseases. Many statistical tools exist and have been used with success in genotype association studies, but success has been limited. Many diseases have proven to rely on multigenic effects or interaction with environment that are not detectable using linear regression on small subsets [15, 34]. Current work [12] is under way to utilize machine learning techniques to uncover more complicated multilocus effects or effects with environmental influences. This thesis is concerned with computational methods to perform genotype association studies using unmatched case control data. We review several of the traditional statistical methods

as well as more modern machine learning competitors. We identify the statistical hypothesis that each algorithm tests and present results on 11 simulated genetic models to judge power and type 1 error for each algorithm. In addition, we introduce a novel use of Alternating Decision Trees that shows promise to scale well to large data sets. We find that decision tree algorithms are suitable for feature detection in many instances, but that they are not able to detect interactions that lack marginal effects. The question of whether interactions without marginal effects should be a major focus of effort is still open. We also find that, unlike several popular machine learning algorithms, different interpretations of ADTrees enables differentiation between association and statistical interaction. We conclude by introducing a tool called SNPdoc for combining the results of genome-wide association studies (GWAS) with biological data stored in online databases.

1.1 A Whirlwind Introduction to Human Genetics

Deoxyribonucleic acid (DNA) is a polymer composed of segments called nucleotides. Each nucleotide consists of a phosphate and sugar residue and multiple nucleotides bond in a line to form the backbone of the macromolecule. The bond is asymmetric, meaning the backbone has a directionality that is exploited when recording genetic sequences. Each nucleotide also contains a single nucleobase from the set adenine (A), cytosine (C), guanine (G) and thymine (T). Two DNA molecules form hydrogen bonds (A with T and C with G) to form the iconic double helix, a structure with high stability that aids in the long term transmission of the macromolecule. In humans, DNA is divided into 46 chromosomes. Twenty-two pairs of chromosomes are nearly exact copies of each other and one copy is contributed by each parent. The last pair consists of either two X chromosomes or an X and a Y.

Another molecule, RNA, forms the messenger between DNA and the proteins

whose construction it encodes. RNA differs from DNA in the composition of the backbone and because thymine is replaced with uracil (U). RNA has the ability to copy a strand of DNA using the same hydrogen bond structure as mentioned previously. It is then transported to ribosomes that perform the task of protein creation. Each triad of bases forms a codon that codes for a single amino acid. Most of the human genome does not code for a protein. The regions that are coding are in genes, which are further divided into exons, which actually code, and introns, which serve other purposes. Interactions between genetically coded proteins and environmental factors produces the phenotype, which is the set of physical and physiological attributes of an individual.

Many of the genetic models described in this thesis utilize the vocabulary of Mendelian gentics. We say an allele is in a dominant relationship with a phenotype if the phenotype occurs with just one copy of the allele. We say it is in an additive relationship with the phenotype if the phenotype's intensity increases with each copy of the allele. Last, we say it is in a recessive relationship if the phenotype occurs only if two copies of the allele are present. While many traits do not follow any of these patterns exactly, they provide a simple means of explaining patterns of genetic expression.

Another common idea is linkage disequilibrium (LD), which is defined as an association between two nearby loci. Formally, two loci are in LD if they are close enough physically that the probability of recombination between the two loci is less then 1/2 and if there is a statistical relationship linking two loci. We will not discuss the specific measures used for LD because they do not form a major focus of this thesis. For more details, see [20].

DNA is incredibly stable: there are regions of the human genome that directly map to regions of DNA in bacteria. However, mutations do occur, and they have

built up over time to form different organisms. Within humans, less than 1/10 of 1% of genetic material differs from other humans. The locations that differ are called single nucleotide polymorphisms (SNPs) and form the basis of the statistical genetics covered in this thesis. SNPs are typically given a unique identifier like $rs10000003$ that can be used to gather information from many sources about the same location (for instance, using SNPdoc). By recording the genetic information at each of the SNPs, it is possible to capture a large portion of the variation among humans, and methods can be applied to gather information about genetic differences that associate with different expressions of a specific phenotype such as disease status. The rest of this section is concerned with the statistical terminology for analyzing and reporting results in computational genetics.

1.2 Categorical Data and Statistical Hypothesis Testing

Any attempt to prove something is so using statistics will run into the problem of induction: it is not possible in general to prove something always occurs by observing it some of the time. Instead, statisticians operate by proposing a hypothesis and then providing the odds of observing the data given that the hypothesis is true. Typically, the hypothesis is the exact opposite of the thing that one suspects. For instance, in genetics, if one has recorded the genotype at a particular locus of a sample from a population that differs in some phenotypic attribute, one might make the hypothesis that the value at that locus does not correlate with the phenotype. This hypothesis is called the null hypothesis, or H_0. One uses statistical techniques to calculate the odds that H_0 is true given that the value for the statistic measured is as extreme or more extreme than was observed. We say a null hypothesis is rejected as significance level α is the probability of observing a test statistic or a more extreme statistic is

less then α.

The most important step in statistical hypothesis testing is clearly identifying the null hypothesis, H_0, and the alternate, H_A. Any assumptions about the population as measured are also important and form part of the testable H_0. In the ideal case the two hypotheses are the only alternatives and $H_0 = \neg H_A$. However, care must be taken to identify the true alternative hypothesis, as it may differ from that which is hoped. Rejecting H_0 means that the explicitely stated H_0 is assumed false or that an assumption is faulty, and neither result is necessarily equivalent to H_A being assumed true. For instance, it is widely known that Sickle-cell disease is a genetic disease found most commonly in individuals of Sub-Saharan African descent. Were one to gather a large sample of humans, genotype them at an array of SNPs and at each locus test H_0 : the locus is not associated with Sickle-cell disease, one would likely find very many loci where H_0 is rejected. This would not imply that H_A : the locus is associated with Sickle-cell disease is true. Another implication might be that the sample was incorrectly gathered: a disease that shows up disproportionally often in a single ethnic group will cause genetic differences between the ethnic group and other humans to appear as associated with the disease. This is an example of a hypothesis that is rejected because of faulty assumptions rather than a false H_0. Much of the remainder of this thesis is concerned with accurately identifying the statistical hypothesis that is tested by each algorithm.

Once H_0 and H_A are clearly defined, one must choose a test to perform and compute its correct test statistic. The test statistic is a summary value that describes some property of the data that is of interest. Each algorithm in this thesis is a particular test that can be used to test a particular H_0 and return a test statistic. If an analytic form for the distribution of the statistic under H_0 is known, one can use that distribution to judge the likelihood of observing a given test statistic. A value

is chosen that divides test statistics into two regions, called the region of acceptance and the critical region. Test statistics in the critical region imply that H_0 is rejected with a probability equivalent to the proportion of test statistics in the critical region. Alternatively, one can use permutation testing [14] to calculate the empirical distribution of a test, and use the empirical result to gauge the likelihood of observing the statistic or a more extreme statistic from the data.

Tests can be compared using several measures of their effectiveness. Power and type I error are two of the more important measures. Power is defined as the probability of rejecting H_0 if it is actually false. Type I error is the probability of rejecting H_0 when it is actually true. A related statistic, positive predictive value, is the probability that a rejected H_0 is actually false:

$$PPV = \frac{\text{Power}}{\text{Power + Type I error}}. \tag{1.1}$$

Machine learning literature has a similar set of terminology, which it is helpful to review. Sensitivity, also called the true positive rate, is equivalent to statistical power: it describes the percentage of time that H_0 is rejected if it is false. Specificity is defined as the probability that a true H_0 will not be rejected. Since in our case the number of instances in which H_0 is false is very small, specificity is always 0.95 or higher, rendering the statistic meaningless. Two terms that are used in the document retrieval setting are precision and recall. Precision is equivalent to positive predictive value, and can be thought of as the percentage of time that a rejected H_0 is actually false. Recall is equivalent to both sensitivity and power. See [22, 28] for more information about measures of algorithm success in document retrieval and in the clinical setting. Since the target users of the machine learning algorithms discussed in this thesis are statisticians, we elect to use that language in the rest of the document.

A common test used in settings where categorical data is observed is based on

a contingency table [1], which allows one to catalogue all possible outcomes and count their occurences. For instance, suppose $n = 1000$ individuals are split into two groups, where one group has a disease (cases) and the other does not (controls). Suppose the genotype is known at a SNP for each individual. If we make H_0 : there is not a dominant relationship between the SNP allele 'a' and the disease, then we can construct the contingency table in Table 1.1.

Table 1.1: Contingency table comparing genotype frequencies using a dominant model. The values a, b, c,and d are the counts of the individuals that fall into each of the four possibilities.

	Genotype AA or Aa	Genotype aa
Cases	a	b
Controls	c	d

Under H_0 and the assumption that individuals are unrelated and come from the same genetic population, the quantity

$$\frac{2n(ad-bc)^2}{(a+c)(b+d)2n^2} \tag{1.2}$$

is asymptotically equivalent to $\chi^2_{1d.f.}$ [20]. The $\chi^2_{1d.f.}$ distribution is equivalent to the square of the standard normal distribution. Therefore, it is possible to test H_0 using a standard and well known test statistic. However, care must be taken in its interpretation, as H_A is not simply that the locus of interest is associated with the disease. Rather, it could be in LD with another locus that is itself associated with the disease, which occurs when two nearby loci have a nonrandom relationship between their expressions.

A contingency table can be used to further understand power and type I error. Consider two tests, A and B which each test hypothesis H_0. In addition, let test A be the gold standard, meaning that its determination is always considered to be accurate. The table in 1.2 compares the two algorithms in a contingency table. Value

a corresponds to the number of independent trials in which both test A and test B accept H_0. Since $A : rej$ is considered equivalent to H_0 being false, $d/(b+d)$ is the proportion of trials in which test B correctly rejected a false H_0 (power). Similarly, $c/(a+c)$ corresponds to the percentage of time that test B rejects a true H_0 (type I error).

Table 1.2: Normalized Contingency table comparing two statistical tests. The values a, b, c,and d are the counts of the individuals that fall into each of the four possibilities.

	A: acc	A: rej
B: acc	a	b
B: rej	c	d

Using the contingency table in Table 1.2, we can determine whether tests A and B have equivalent power. Under H_0 that the two tests have the same null hypothesis and that their power is equivalent, it is possible [1] to calculate that

$$\frac{(b-c)^2}{b+c} \approx \chi^2_{1d.f.}. \tag{1.3}$$

We will use this test, called McNemar's test, to compare the power of algorithms in Chapter 4.

Chapter 2: Alternating Decision Trees

This chapter introduces the ADTree machine learning algorithm as well as some of the customizations and methods of interpretation that we have developed specifically for SNP data. We start by briefly considering decision trees and Adaboost, then their combination in ADTrees. Finally, we introduce and justify the use of bagging for SNP identification, including a novel interpretation method that focuses on sets of related SNPs in multiple ADTrees.

The algorithms introduced in this section have been written in C++ by the author specifically for SNP data in the two columns per SNP "linkage" format. They have been designed to run in parallel in a shared memory environment using OpenMP. Another popular general purpose implementation is available in Weka [16], which is written in Java.

2.1 ADTrees

Consider a training set

$$\mathcal{T} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\},$$

where $\mathbf{x}_i$ is an instance with classification $y_i \in \{-1, 1\}$. Elements of the vector $\mathbf{x}_i$ denote attributes, which are measurable qualities of an instance (in this case, SNPs.)

One popular family of classifying learners is the recursive partitioning, or decision tree, class of algorithms. A recursive partitioner builds a set of recursive decisions, often depicted in a tree or as conjunctions of rules, that subdivides input data into progressively smaller subsets that are eventually associated with a classification. Classic examples of recursive partitioners are the CART [5] algorithm and C4.5 or the

commercially available C5.0 algorithm [38]. ADTrees [13] are similiar in appearance to classic recursive partitioning learners, but they differ in their interpretation and construction. In particular, they provide some of the structured knowledge of a tree while avoiding the potentially large and complicated structure of a binary decision tree. In this section, we will develop the ADTree algorithm by focusing first on boosting, which is a technique for improving the performance of small learners, followed by the extension of boosting to decision trees in the ADTree algorithm. We conclude with a discussion of bagging to reduce the effect of noise.

Boosting combines a set of learners into a voting committee to often achieve improvements in classification quality. A boosting algorithm iteratively builds simple learners that are combined into a meta-learner. Each simple learner is given a weighted vote. In addition, boosting algorithms assign a distribution to the instance set that is updated each iteration, which changes the weight of each instance in the set. Since the distribution of instances is over a finite set, it is equivalent to weighting the instances such that the sum of the weights is 1. A basic boosting algorithm called AdaBoost [38] is presented in Algorithm 1.

Input: Simple Learner $\mathcal{S}$
Data set $\mathcal{D}$
Number of Iterations T
Result: Set of simple learners with weighted votes. Final classification by weighted majority vote.

$W_1(i) = 1/m$ `// m is size of` $\mathcal{D}$
for $t = 1 \ldots T$ **do**
  $h_t = \mathcal{S}(\mathcal{D}, W)$ `// Train simple learner using` $\mathcal{D}$ `and` $\mathcal{W}$
  $\epsilon_t = \Pr_{i \sim W_i}(h_t(\mathbf{x}_i) \neq y_i)$ `// Calculate error from learner`
  $\alpha_t = 1/2 \ln\left(\frac{\epsilon_t}{1-\epsilon_t}\right)$ `// Weight the learner`
  $D_{t+1} = \frac{D_t \exp(-\alpha_t y_i h_t(\mathbf{x}_i))}{Z_t}$ `// Weight the instances and normalize`
end

Algorithm 1: AdaBoost Algorithm

The simpler learner is treated as a black box in the boosting algorithm, though

typical candidates are simple and do not significantly outperform chance. The authors in [18] demonstrated that the complexity of the simple learner is sometimes inversely proportional to the final classification quality, so single node trees (stumps) are typical simple learners. Another assumption is that the distribution W_t is used in the simple learner either directly or as a resampling criteria.

The error in each step is calculated as a weighted sum of all misclassified individuals, giving larger positive weight to a better classifier. Each instance is given a weight of

$$D_t(i)\frac{1-\epsilon_t}{2\epsilon}\exp(-y_i h_t(\mathbf{x}_i)), \tag{2.1}$$

which is then normalized so the weights form a distribution.

Two voting procedures are intertwined in AdaBoost: simple learners are given a weighted vote in the final classifier based on their predictive success, and each instance of the training data is given a weighted voice in the next simple learner. Two scenarios exist for Equation (2.1). Suppose $y_i h_t(\mathbf{x}_i) = -1$, i.e. the instance is misclassified. The instance is given greater weight ($e^{x>0}$) with magnitude ultimately determined by the efficacy of the simple learner. If $y_i h_t(\mathbf{x}_i) = 1$ the instance is correctly classified and given less weight. As a result, instances that are difficult to classify accrue the most weight while instances that are always correctly classified are gradually ignored.

A concern with boosted algorithms is that later iterations will focus on special cases at the potential expense of incorrectly classifying instances that fit the overall class definition more closely. In the case of genetic data, common forms of noise include admixture and mistakes in genotyping. We will investigate this problem in more detail after introducing ADTrees.

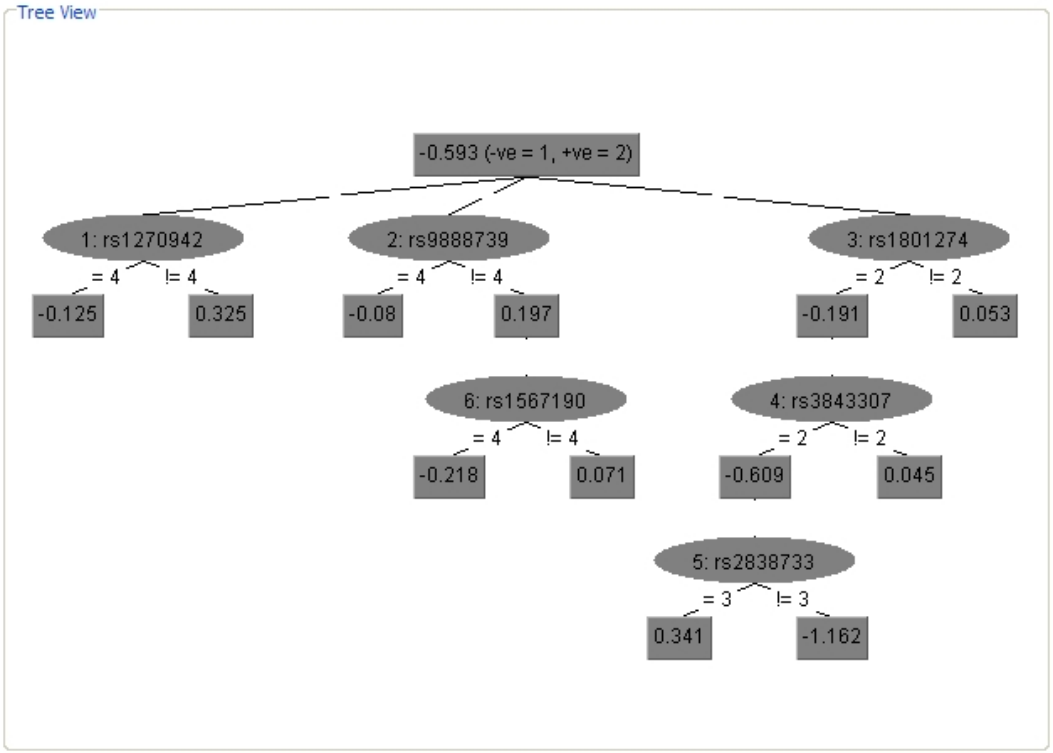

Figure 2.1: An Alternating Decision Tree as produced by Weka. Prediction nodes are ovals while decision nodes are square. The number in front of each SNP is the order in which nodes were added. This tree was built using categorical rather than numerical data, so decisions are of the form "equals n" and "not equals n." To classify a given individual, one would follow each path with a decision that is true and sum the scores. The sign of the final scores is the classification.

2.1.1 The ADTree Algorithm

Alternating decision trees (ADTrees) provide an extra layer of structure to the set of weak learners derived in the boosting algorithm. ADTrees consist of alternating layers of decision and prediction nodes. In Figure 2.1, prediction nodes are oval while decision nodes are square. Prediction nodes identify an attribute and contain two leaves representing a partition of the attribute domain. When evaluating an instance, only one path from a prediction node can be followed. Decision nodes contain a number called a score and an arbitrary number of prediction node children. When evaluating an instance, all children from decision nodes are followed in parallel by an instance reaching the node. ADTrees start with a decision node, so traversal of an ADTree results in a set of paths through the tree such that one path is taken at each prediction node that is reached. An instance is given a score by summing the scores of all decision nodes that are traversed by that instance.

The ADTree algorithm, introduced in [13], produces a set $\mathcal{P} = \{P_i\}$ of preconditions and a set $\mathcal{R} = \{R_i\}$ of rules. A single rule consists of a simple conditional involving a precondition, a test condition c_i and a set of signed numerical predictions p_1 and p_2. Its form can be found in Algorithm 2, and final classification is given by

$$C(\mathbf{x}) = sign(r_0 + \sum_{R_i \in \mathcal{R}} R_i(\mathbf{x})), \tag{2.2}$$

where r_0 is the decision value in the head node. Test conditions are single predicates over instances limited in this work to statements of the form

$$\text{attribute } n \ \Theta \text{ value } v \tag{2.3}$$

for comparison operator Θ. For instance, a condition might be SNP $rs1234 \geq 3$. The set of all test conditions is labeled $\mathcal{C}$. Preconditions are conjunctions of conditions and negations of conditions.

Input: Precondition $P \in \mathcal{P}$, Condition $c_1 \in \mathcal{C}$, Scalars $p_1, p_2 \in \mathbb{R}$
Result: Number (either p_1 or p_2 (or 0) denoting a single prediction)
if P **then**
    if c_i **then**
        return p_1
    else
        return p_2
    end
else
    return 0
end

Algorithm 2: Basic ADTree rule

An iteration t of the ADTree algorithm produces a new rule R_{t+1} using a condition c_t and a precondition P_i, which can be the result of any previous iteration. In iteration one, set $P_1 = \{true\}$. An iteration produces two new preconditions $P_i \wedge c_t$ and $P_i \wedge \neg c_t$. A significant feature of the ADTree algorithm is that it allows for rule R_{t+1} to use any previous precondition. In contrast to traditional binary decision trees, where trees grow only from the leaves of previous iterations, ADTrees can take non-binary forms. Preconditions with shared prefixes form a tree structure, which sets ADTree apart from AdaBoost.

ADTrees contain several other improvements over typical recursive partitioning trees as well as an improved interpretation method over simple AdaBoost. Among the advantages are:

- Parallel paths can signify sets of attributes that independently classify the data. The authors in [21] have noted that this plays a particularly important role when multiple noninteracting genetic markers influence disease.

- Traditional trees often have to be much larger than ADTrees to get the same performance [12]. As a large tree is created, data is subdivided into smaller sets leading to potential overfitting.

- ADTrees provide multiple levels of interpretation [13, 29]:
  - The entire tree can be treated as a whole.
  - A single rule and its score (a single path or sub path in the tree) can be treated alone and given a confidence score.
  - Each prediction-decision-decision triad can be considered as a vote by the single attribute in the prediction node. The score allows for a measure of confidence in the prediction.
- Unlike traditional AdaBoost, there is a relationship between simple learners based on the prefixes of preconditions.

ADTrees also present a few unique challenges:

- The algorithm's complexity is $O(k^2 n)$, where k is the number of iterations and n is the number of attributes (SNPs) because each iteration must consider every previous precondition. (See [29].) To some degree, the size of the trees under consideration ($k < 20$) obviates this concern. Smaller trees are the result of the improved accuracy compared to a binary tree of the same size and our goal of eventual use of ADTrees in feature detection, discussed below.
- With multiple levels of interpretation comes multiple levels of complexity. Which is the correct interpretation for SNP data, or should we examine multiple levels?
- As with all tree based methods, at least one SNP in an interacting set must display a strong enough marginal effect to make it into the tree. Interactions with no marginal effect will be missed [12, 23].
- As with AdaBoost, noisy or mislabeled data can have a large negative effect on performance. This will form the subject of the next section.

In the next section, we consider a feature detection algorithm using ADTrees that makes use of the richer structure than binary trees mentioned above. In addition, concern over the ADTree algorithm's quadratic compexity in the size of the tree is alleviated because the feature algorithm favors smaller trees.

2.2 Bagging ADTrees to Cancel Noise Effects

Mislabeled instances or outliers that suffer from admixture or erroneous genotyping will be misclassified by most of the simple learners in a boosted algorithm, leading later learners to weight poorly representative instances too much. The effect is tempered in part by the fact that a learner that is trained on outliers at the expense of the main body of instances will have less weight in the final vote. A more stringent correction is to use bagging (Bootstrap AGGregatING) to reduce the influence of noise (see [2, 29]).

Bagging works by producing a set of bootstrap samples (sample instances with replacement) from the original data and building a classifier on each sample. A final classifier is provided by a vote, generally unweighted, of the bagged classifiers. In terms of classifier accuracy, [2] has demonstrated that bagging works best for unstable predictors where different bootstrap samples produce very different classifiers. In the context of SNP feature selection, instability is provided by the complex interaction of genetic data, by the presence of noise, and by the underlying tree classification which is greedy in the addition of new features. Different bootstrap samples will contain different over and under representations of disease individuals causing the over or under expression of different sets of interacting SNPs.

Input: Training data $\mathcal{T}$ where $T_i = (\mathbf{x}_i, y_i)$
Procedure $W_+(C)$ calculating total weight of all instances satisfying condition (pre- or test) that are in positive case.
Procedure $W_-(C)$ calculating total weight of all instances satisfying condition (pre- or test) that are in negative case.
Result: Rule set $\mathcal{R} = \{R_i\}$ for $1 \leq i \leq T$.

$W(i) = 1$ `// Give equal initial weight to each instance.`
$\mathcal{C}$ = set of all possible test conditions
$a = 1/2 \ln \frac{W_+(\text{true})}{W_-(\text{true})}$
R_1 = rule whose precondition and test condition are both "true" with prediction p_1.
$W(i) = W(i)e^{(-ay_i)}$ `// Reweight training data so elements of less common class weighted more.`
for $i = 2 \ldots T$ **do**
p, c = Values that minimize

$$Z_t(p_j, c_k) = 2\left(\sqrt{W_+(p_j \wedge c_k)W_-(p_j \wedge c_k)} + \sqrt{W_+(p_j \wedge \neg c_k)W_-(p_j \wedge \neg c_k)}\right) + W(\neg p_j)$$

for $p_j \in \mathcal{P}$ $c_k \in \mathcal{C}$ and $W = W_+ + W_-$.
// NOTE: The above minimization is over a finite domain.
$\mathcal{P} = p \wedge c + p \wedge \neg c + \mathcal{P}$
$a_1 = \frac{1}{2}\frac{W_+(p \wedge c)+1}{W_-(p \wedge c)+1}$
$a_2 = \frac{1}{2}\frac{W_+(p \wedge \neg c)+1}{W_-(p \wedge \neg c)+1}$
Rule R_{t+1} given precondition p, condition c and weights a_1, a_2 respectively.
$W(i) = W(i)e^{-R_{t+1}(\mathbf{x}_i, y_i)}$
end

Algorithm 3: Basic ADTree Algorithm

2.2.1 Bagging ADTrees for Feature Selection

Features are defined as individual attribute-value pairs or sets of attribute-value pairs. Feature selection is the process of identifying features that best classify the data. In the context of GWA studies, features are interpreted as sets of SNPs and alleles that best classify instances as case or control. ADTrees and many other methods are designed to classify, but they must first identify features in the data that are useful for classification. Bagging can be used to find common features across multiple ADTrees, but care must be taken to understand features in the context of multiple trees. When building a single ADTree, features are defined as attributes from the data that are selected for their ability to minimize the quantity $Z(p, c)$ in Algorithm 3. Equivalently, features are attributes that minimize uncertainty given the data as weighted by previous steps in the algorithm. When examining n bagged ADTrees, the sample is a set of trees that were constructed on different data sets. Therefore, features in terms of bagging are defined as structural or numerical properties of the trees, not the original data.

Algorithm 4 focuses on structural features of trees in the form of paths. A path is defined as a single set of alternating prediction-decision-prediction nodes through the tree that starts at the root and ends in a leaf node. Paths are equivalent to elements of the rule set $\mathcal{R}$. The result of Algorithm 4 is a list of paths that were in the most ADTrees in an iteration, with the threshold of definition controlled by a parameter. Subtrees, which are defined as a portion of a tree extending from some decision node, are another structural feature that is considered.

Features in the sample of n trees can be associated with features in the original data. Algorithm 3 is designed to extract features in the form of SNPs and sets of SNPs that serve the purpose of classifying instances of the bootstrapped data set. Features that appear in multiple bootstrap samples will show up in multiple ADTrees

Input: Training data $\mathcal{T}$
Number of bags n
Threshold for bags a, b where $1 < a, b, \leq n$
Result: finalPaths = Set of ADPaths

```
// Parameter List
numNodes = 10
bagInputSize = 100%
minTrees = a
while True do
    Remove all SNPs in all paths in finalPaths from input file
    Build n data sets using sample with replacement
    Build an ADTree using algorithm 3 on each data set
    allPaths = array of all unique paths in all trees
    for each path p in allPaths do
        if p is in more than minTrees trees then
            Add p to finalPaths
        else
            break while
        end
    end
end
```

Algorithm 4: Bagged ADTree Algorithm on full paths.

and will be detected by Algorithm 4. A fundamental assumption of our work is that those features that are important in multiple bagged sets are more likely to be important in the original set, as they were detected multiple times. Therefore, the bagging algorithm measures the importance of features in the original data by measuring the number of bootstrap samples in which they are important. Selection of the number of trees is motivated by the desired accuracy and by the desired threshold level of importance desired of the features. A greater number of trees gives a more nuanced degree of precision while allowing for detection of less significant features. In order to detect features that appear in, say, 5% of the trees, one would need more trees than required to find features that appear in 20% of trees. We use 100 trees in the experiments that follow, which allows a reasonable degree of precision but is still computationally feasible. Empirical testing demonstrated that several subpaths tend to appear in 5 to 7 trees, which would imply that one can set a threshold of acceptance in that range for 100 trees.

There are several advantages and disadvantages to this type of analysis. A major advantage is that the numerical output consisting of the iteration in which a path was found and the number of trees containing the path are easily interpreted. Since SNPs are removed from consideration if they are in a path that is in enough trees, all subsequent paths must be understood as features only when the prior paths are removed. Also, a feature's importance (here feature is structure in an ADTree) is denoted by the percentage of trees in which it occurs. However, the drawback to this kind of analysis is a tendency to identify small paths involving fewer SNPs rather than larger paths. For instance, if analysis only considers full paths (end at leaves) then two paths that are equivalent except for the last SNP will not be considered as equivalent and will not be removed. Considering subpaths rather than full paths in the bagging algorithm corrects this problem. We present results using both full paths and subpaths and conclude that subpaths are a more powerful method.

Another disadvantage is that SNPs removed in round i are no longer available to form potentially important paths in later rounds. Investigation of the severity of this disadvantage was not performed, but it does not appear to merit immediate consideration. In order for this problem to exist, a SNP must be in two interacting and overlapping sets of SNPs. Unless power to detect one subset far outweights power to detect the other, we expect them both to show up as features.

2.3 Interpretation

Interpretation of the output from Algorithm 4 is straightforward: those features which show up in multiple of the bagged trees are the most significant features. A single ADTree is more difficult to interpret for two reasons. First, the tree is grown to a large size and may include both truly associated SNPs and noise SNPs. Determining how large build a tree to capture only ground truth SNPs is impossible *apriori*. Second, once the tree has been grown, it is difficult to judge the degree of confidence that should be associated with the tree or with any subtrees. One measure that is mentioned in the literature is the size of the confidence prediction, but this was a poor descriptor of the ground truth in initial investigations. A better measure is the predictive performance of the tree as measured either on the data set itself or through 10-fold cross validation (CV.) Cross validation splits the input into n random subsets, then performs the algorithm n times leaving one of the subsets out in each model. The subset that was left out is classified on the model built on the rest of the data, and ability to classify the "out of bag" individuals is recorded. While stopping criteria do not form a part of this body of work, the problem remains largely open for ADTrees. We chose to halt production at 10 prediction nodes as this includes 5% of the total input. An alternative that deserves investigation is halting construction of the tree when the predictive success as measured by Cohen's Kappa [37] stops

increasing. Predictive success can be used in two related ways: first, construction can be halted when the success ceases to improve. Alternatively, construction can be halted based on size and permutation testing on trees of a particular size can be used by permuting the phenotype and measuring predictive success. In this case, the null hypothesis is that trees of the given size are not predictive of phenotype, and the null hypothesis is rejected if prediction using the test tree is better than prediction in a sufficient number of the trees built on permuted data depending on the rejection criteria in use.

Chapter 3: Other Algorithms and Testing Procedures

Statistical hypothesis testing requires that the null hypothesis be explicitly stated. In this chapter, we introduce the algorithms against which ADTrees and Bagged ADTrees are tested. The purpose is twofold: we introduce the algorithms and the justification for each and we make clear what hypothesis each algorithm is testing. We also introduce the genetic models used to test the algorithms, and we close with a discussion of the criteria for rejecting H_0.

3.1 Logistic Regression

Logistic regression is one of the workhorses of statistical genetics, and we will test four variations of the general framework. In particular, we will test for association between a single SNP and a phenotype, for interactions between SNPs with an expected marginal effect, for pure interaction between SNPs, and for the presence of a set of associated SNPs using a stepwise procedure.

3.1.1 Overview

Regression models attempt to predict the value of a response variable using available explanatory variables. In the literature on regression, explanatory variables are also called independent variables and response variables are also called dependent variables. We will be using the terms SNP and phenotype as synonyms for explanatory or independent variables and response or dependent variables, respectively. Linear regression attempts to model the dependent variable y using independent variable x

by the equation

$$y = \alpha + \beta_1 x + \epsilon. \tag{3.1}$$

The ϵ term is the error term and is expected to be randomly distributed with mean 0 though the distribution is application dependent. If y is a boolean variable, for instance, case or control, the linear regression model presents problems. First, y takes values in the set $\{0, 1\}$, and almost all values of x produce values for y outside that range. Even if we allow for $y \in [0, 1]$ and use rounding, most x miss the unit interval. Second, there is often a nonlinear relationship between x and y when y is boolean [1, ch: 4]. A unit change in x should produce a greater change when y is near 0 or 1 than when y is near 1/2. The solution provided by logistic regression is to perform a transformation of the probability space $[0, 1]$ to the entire real line.

In order to introduce logistic regression, it is helpful to define a few quantities. Assuming that the response variable is always case or control, label case as 1 and control as 0. Let $E(y) = P(y = 1)$ be the expected value of y. In order to suggest the dependence of y on the explanatory variables $x_1, \ldots x_n$ it is common to denote $E(y)$ as $\pi(x)$ where $x \in \mathbb{R}^n$ is the vector of explanatory variables. Rather than fit a line to $\pi(x)$ directly, the log odds, or logit transform

$$logit(p) = \log \frac{p}{1-p} \tag{3.2}$$

is applied. The logit maps $(0, 1) \to (-\infty, \infty)$, which solves the first problem above. In response to the second problem, the logit is also more sensitive near its domain boundaries. Logistic regression predicts the logit by the equation

$$\frac{\pi(x)}{1-\pi(x)} = exp(\alpha + \beta x) \tag{3.3}$$

or

$$logit(\pi(x)) = \log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \alpha + \beta x \tag{3.4}$$

in the case that x is a single variable.

Consider the null hypothesis that the scalar variable x has no relationship to y. Changes in x should have no bearing on the value of y. In equation (3.4) no relationship corresponds to $\beta = 0$. Estimates for β are asymptotically normal, so the Wald test of significance of $H_0 : \beta_0 = 0$ applies [1]. The presence of a test of significance and the capability of dealing with multiple covariates have made logistic regression a gold standard algorithm.

Statistical interaction between a pair of variables x_1 and x_2 is defined as deviation from additivity in the relationship between the two variables. In the logistic equation

$$logit(\pi(x)) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1 x_2, \tag{3.5}$$

the first order terms correspond to an additive relationship. If β_3 is nonzero then a nonadditive relationship is said to exist: in this case it is a multiplicative relationship. Other terms, for instance $\beta_n x_1^2 x_2^2$ would capture different types of interaction. In Chapter 4 we will refer to the LR test of two SNP interaction as LR2.

Two modifications to the basic logistic regression algorithm are included in the study. First is a marginal test of interaction. A test for marginal effect in each SNP is performed using single-SNP logistic regression (equation (3.4)) and a low threshold of significance ($p = 0.2$) for rejecting H_0 of no single-SNP association. SNPs that meet this threshold are then tested for two way interactions using logistic regression of the form in (3.5). The null hypothesis in the final test is that there is no interaction between the two SNPs x_1 and x_2. All pairs that are identified as under the alternative will also have at least a slight marginal effect, so power is expected to be reduced compared to the full two way logistic test for interaction. The severity of the power reduction depends on one's worldview. If most genetic interactions are between SNPs with a marginal effect then power will not be significantly reduced. However, if the primary focus is on detection of pure interactions (i.e. interactions without a main

effect) then power will be reduced to near 0.

The last logistic regression model under consideration, called stepwise regression [19], does not make assumptions about the size of models (either one or two SNPs above). Stepwise regression is an iterative algorithm that constructs an LR model by successive adding and removal of explanatory variables provided they meet a minimum p-value threshold. At each step, a LR model is constructed using all of the variables already in the model and a single variable that has not been added. The out of model variable with the lowest p-value is added to the model provided it meets a minimum entrance criteria. If a variable is added, LR is performed on the entire model. Any SNP that does not meet a minimum stay criteria is removed from the model. The process halts when an iteration fails to alter the model. The null hypothesis is different than those previously considered because the entire set of explanatory variables is included rather than just one or two explanatory variables at a time. In the previous three LR variants, the null hypothesis was that there is no association between a single independent variable or pair of independent variables and the dependent variable. In stepwise LR, the null hypothesis is that there is no variable or set of variables that explains the dependent variable.

3.1.2 Implementation

SNPGWA [32] is a tool written at Wake Forest University to perform single SNP analysis. Among the algorithms it implements is logistic regression of the form in (3.4). Intertwolog is a tool written at Wake Forest University to perform logistic regression and significance testing on equations of the form (3.5). We use these as our primary single and double locus logistic regression engines. Marginal logistic regression is performed using a Perl script and both programs. Stepwise regression uses SAS with entry and stay thresholds set at $p = 0.01$. Since stepwise LR was not a major focus of this thesis, we did not investigate a range of thresholds. The threshold

$p = 0.01$ threshold appears to produce power and type I error below expectation for single SNP models.

3.1.3 Advantages and Disadvantages

As a purely statistical tool, logistic regression is one of the most widely used tools for genetic association studies. It allows natural extensions to deal with covariates and, as was mentioned previously, provides a robust theoretical foundation for hypothesis testing. In addition, it is possible to test for statistical interaction directly rather than simply for association.

The two biggest problems with logistic regression are computational complexity and model specificity. Logistic regression in multiple variables suffers from the curse of dimensionality. A test for n-way interaction requires 2^n terms in the regression equation. In addition, the computational complexity of $\Theta(n^k)$ for a test of k way interaction on n SNPs makes computation costly for GWAS size data sets. Logistic regression also requires that the model be specified ahead of time. For instance, the null hypothesis in (3.5) that there is no multiplicative interaction is not to be mistaken with the null hypothesis of no interaction. For this reason, interaction testing with logistic regression will suffer reduced power [14, 31]. Logistic regression on one SNP assumes an additive model, and typical implementations will also test for dominant and recessive association.

Stepwise logistic regression has its own tradeoffs. It makes less of an assumption about the size of final models without requiring that large exhaustive tests be performed. However, its p-value is difficult to interpret because of the iterative nature of the algorithm. In addition, the method detects association rather than interaction. The final model, $logit(y) = \beta_1 \mathrm{SNP}_1 + \beta_2 \mathrm{SNP}_2 + \cdots + \beta_n \mathrm{SNP}_n$, does not include any interaction terms.

3.2 Multi-factor Dimensionality Reduction

Multi-factor dimensionality reduction (MDR) [15, 24, 25, 30, 31] is a variant of combinatorial partitioning [27] that was designed specifically for genetic data. The algorithm is in a mature state with an open source Java implementation [15].

3.2.1 Overview

As a combinatorial partitioning algorithm [27], MDR works by identifying sets of variables that maximize the ratio of counts of the two classes. Unlike logistic regression, no assumptions are made about the specific model contained in the data.

MDR works iteratively on all size n subsets of the input as well as on values of n between 1 and a user specified upper bound. A single step in the MDR algorithm uses a set of n polymorphisms chosen from the entire input set. The ratio of cases to controls is recorded for each of 3^n possible multilocus genotypes. Ratios greater than a certain threshold denote genotypes that consist of mostly cases and are denoted high risk genotypes. MDR reduces the dimensionality of the original problem from n polymorphisms to a single variable that takes the values "high risk" and "low risk." Finally, prediction error is calculated on the entire data set as well as by 10-fold cross-validation. The former is used to rank all models of a given size, while the latter is used later for significance testing.

MDR is computed for each set of n polymorphisms and a final $n-$way model is selected that maximizes the risk ratio across all high risk genotypes. The original paper claims that the model with the highest risk ratio also has the lowest classification error if the original set is reclassified using the best model [31]. If models over more than one size are considered, the overall best model is selected as the model that maximizes the risk ratio among the top models at each size.

Significance testing is performed on the best model in two ways. The first test is

of H_0 that the best model will not be consistent across ten CV folds. A model built on data that contains no relationship to the phenotype should produce a random best model that differs across cross-validation folds. Significance is reported as a z-score over the distribution of consistencies derived from 1000 empirical permutations that were precomputed in the software. Association and interaction are indistinguishable in this test. Permutation testing is also performed to test H_0 of no interaction. With the latest version of MDR [15], an explicit test of interaction is available that performs stratified permutation of case and control. Rather than permuting phenotype, the data is sorted by phenotype and SNPs in each class (case and control) are permuted. Any marginal relationships with phenotype are preserved while interactions are removed. A point of caution should be made regarding the significance testing in MDR: both hypotheses are for the best model only. A model could exist under H_A, but not be selected as the best model by the algorithm. The importance of this distinction arises when the best model fails to reject the null hypothesis, which may (falsely) lead one to infer that no association exists in the data set.

A third interpretation of MDR is included in our results which has the ability to consider the performance of models that do not rank highest in classification accuracy. The MDR algorithm returns a fitness landscape which compares all models of a given size, and we use that rank to judge whether a model was discovered. Details are in Section 3.6.1.

3.2.2 Implementation

The Java implementation (MDR version 2.0, beta 6, 7) is designed to allow full utilization of multiple cores in a shared memory environment for computing the initial MDR but not for the much longer permutation testing phase. A random seed can be given to the permutation testing engine, allowing distribution across multiple processors if necessary. Along with the cross-validation statistics for the top model, the

"fitness landscape" of classification accuracy for all $n-$way models is also recorded. The only parameter required by the model is the threshold of risk ratio that denotes a high risk genotype. The threshold is important as it tunes the amount of influence of moderately high versus very high risk genotypes. Explicit mention of this value is not made in the literature, but it is assumed that high risk as defined as having strictly more cases than controls. In order to speed computation, we chose to analyze only one, two and three way models.

3.2.3 Advantages and Disadvantages

MDR was designed to provide several advantages over traditional methods, primarily logistic regression. The primary advantage is listed as simultaneous feature selection and characterization provided by the listing of most accurate polymorphism sets along with the specific high risk genotypes. A related advantage is that MDR is non-parametric and makes no prior assumptions about the model type. The authors also list resistance to false positives due to utilization of cross-validation in final model selection as a potential advantage.

Several disadvantages were listed by the original authors as well as by subsequent reviews. Computational complexity and model interpretability are important concerns. Exhaustive testing of $n-$way models for even medium sized n is prohibitive, leading to the need for heuristic or Monte-Carlo techniques. Currently, an exploratory data analysis technique [35] and random searching of a specified number of models or for a specified amount of time are supported by the software. Random searching can lead to erratic or unrepeatable results if an insufficient number of iterations are used. Another important concern is the handling of unbalanced data, which leads to bias in the computation of ratios. Weighting schemes have been offered as a potential solution to this problem. The data under consideration in this thesis is balanced and does not raise this concern.

Another potential problem deserves further mention as it will be a target of further investigation. The CV test of significance for the final model is performed under the null hypothesis of no association at all between the n polymorphisms in the model and the response variable. Thus, the test is unable to differentiate between an effect that occurs as a result of an interaction versus a main effect on one or more of the polymorphisms in the model [12, 30]. This is particularly salient as models with a larger number of polymorphisms will often provide greater accuracy due to greater complexity, despite the fact that they are hiding a rather simple main effect in one or more of the variables. Recently, in [15] the claim was made that linear additive effects no longer confound the test for explicit interaction provided the classification permutation test is used. Both permutation tests suffer from the inability to consider more than a single top model, which may prove to be too narrow in a noisy data environment. More details on the analysis of all three MDR variants is available in 3.6.1.

3.3 Support Vector Machines

Support vector machines have been utilized successfully in a broad number of fields. See [8, 10] for an overview of past successes. Models using the entire set of SNPs are impractical due to the curse of dimensionality, so the form of SVM considered here will involve exhaustive search over smaller sets of SNPs.

3.3.1 Overview

Suppose that data are given in the form $\{(x_i, y_i\}$ such that $x_i \in \mathbb{R}^n$ and $y_i \in \{-1, 1\}$. A linear support vector machine will attempt to find a hyperplane in $\mathbb{R}^n$ that maximally separates the two data classes. A hyperplane can be defined as a vector w and scalar b such that $w^T x = b$. The maximally separating hyperplane, if one exists, is the

minimizer of the following quadratic program [10]:

$$\min_{w,b} \frac{1}{2} w^T w \tag{3.6}$$
$$y_i(w^T x_i - b) \geq 1 \text{ for all } x_i, y_i.$$

In many cases, no separating hyperplane exists, and a relaxed version of the program is considered:

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_i \xi_i \tag{3.7}$$
$$y_i(w^T x_i - b) \geq 1 - \xi_i \text{ for all } x_i, y_i.$$

If the boundary between two classes of data is nonlinear then a transformation $\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is introduced such that the data $\{\phi(x_i)\}$ is linearly separable in the transformed space. The hyperplane is found in the same way, except that $\phi(x)$ is used in place of x :

$$\min_{w,b} \frac{1}{2} w^T w + C \sum_i \xi_i \tag{3.8}$$
$$y_i(w^T \phi(x_i) - b) \geq 1 - \xi_i \text{ for all } x_i, y_i.$$

3.3.2 Implementation

The support vector machine approach utilized in this study utilizes exhaustive search over low order sets of SNPs under a radial kernel function, which was shown by [8] to have better predictive success than a linear kernel. LIBSVM [7] is used as the SVM engine and control software is implemented in Perl. We chose to use a default parameterization, which replicated published results in [8].

Unbalanced input sets can produce bias in prediction error using normal SVMs, so the authors introduce a penalized SVM formulation

$$\min_{w,b} \frac{1}{2} w^T w + C_+ \sum_{y_i=1} \xi_i + C_- \sum_{y_i=-1} \xi_i \tag{3.9}$$
$$y_i(w^T x_i - b) \geq 1 - \xi_i \text{ for all } x_i, y_i, \xi_i.$$

Sensitivity and specificity can be balanced by using different penalty parameters for the two classes. While there is no analytical method to calculate optimal C_+ and C_-, the authors of [8] use a bisection search to approximate appropriate penalty parameters.

3.3.3 Advantages and Disadvantages

SVMs were shown by [8] to identify a 2 SNP interaction model with no main effects in the presence of several kinds of noise, including genotyping error, genetic heterogeneity, phenocopy, and non-differential missing data. Theoretical justification for the improvement in success is provided by the fact that SVMs operate in the dense space in which data is embedded rather than on the data alone. The SVM classifier with recursive feature elimination (RFE) also demonstrated ability to identify models with at least 4 SNPs better than MDR.

Several potential problems exist with the support vector system as considered. An important problem in the literature is confusion over the use of SVMs to test for interaction. Chen et al. define SNP-SNP interaction as, "genotypic combinations of SNPs that are associated with disease status" [8]. Interaction is often defined as statistical epistasis rather than simply sets of associated SNPs because a set of SNPs containing several uncorrelated main effects will appear to be interacting under this definition. See Section 3.6 for more information. The inability to distinguish between interaction as defined in the paper and the traditional definition of statistical interaction has the potential to adversely impact the power of SVMs as well as to produce unacceptably elevated type I error.

3.4 Random Forest

Random Forests (RFs) were introduced by Breiman in [3] as an extension to simple tree methods. They were used for SNP analysis by [6] and the exposition in the following subsections largely follows this approach. Another study of RFs for association studies can be found in [33].

3.4.1 Overview

A random forest is a collection of trees that include structural variations produced by changes in the input data and tree structure. For a sample with n individuals, each tree is grown on a bootstrap sample of size n. The individuals left out of a bootstrap sample are called the out-of-bag set and are used later as a test set. Furthermore, at each split in the tree, a random subset of SNPs are available as potential splits. Since the globally optimal split is not available to some trees, it is possible that some trees will choose less optimal but still important split points. Trees are grown to their full extent and prediction values (case or control) are given to each leaf by majority vote with rare ties being decided as controls. Prediction using a random forest is performed by averaging the prediction over trees in which a sample was out-of-bag.

Importance of a SNP is defined as the change in prediction confidence for out-of-bag individuals when the SNP's value is and is not permuted. For two class problems like those under consideration, the margin is defined as the difference between the proportion of votes for the true class and the proportion of votes for the other class. Define

$$T_i = \sum_{j=1}^{T} t_{ij} \tag{3.10}$$

where $t_{ij} = 1$ if individual i is out-of-bag in tree j and $t_{ij} = 0$ otherwise. For tree j, the vote $V_j(x_i)$ is the value of the leaf node that x_i reaches in the tree. The margin calculation over two classes is given as the average number of correct votes over the

trees in which an individual is out-of-bag:

$$mg(x_i, y_i) = \frac{1}{T_i} \sum_{j=1}^{T} \chi_{V_j(x_i)=y_i} t_{ij} - 1. \tag{3.11}$$

The authors in [6] hypothesize that a SNP that is important for prediction will be in many trees and will be near the root of those trees. If the value for that SNP is permuted among the out-of-bag SNPs, the margin is expected to decrease because an important piece of predictive information is removed. The test null hypothesis of no association between a SNP and a phenotype is tested by permutation testing on the change in margin using a one-tailed t-test. Joint importance of SNPs is performed by jointly permuting the two SNPs among out-of-bag individuals.

Random Forest is the closest algorithm to the bagged ADTree algorithm introduced in Chapter 2. They both utilize a set of trees built on bootstrapped data. However, a few differences exist. First, ADTrees and the CART algorithm used by RF differ substantially in the structure of the trees created. Second, bagged ADTrees uses feature replication across trees to rank features while RF uses classification confidence. RF must test each feature independently, which increases runtime complexity. In addition, bagged ADTrees can be used to distinguish interaction from association.

3.4.2 Implementation

Version 4 of the Random Forest implementation by [4] was used in [6]. We elected to use the updated version 5, which appears not to change the results for these tests. In both versions, the CART algorithm [5] is used to grow trees.

Several parameters are important for correct operation of the RF algorithm, and we largely follow the example set in [6]. Classification accuracy and convergence rate both depend on the number of SNPs that are considered at each split. Splitting over more SNPs is expected to increase prediction accuracy while slowing convergence. While [4] warns not to use the default size of the square root of the number of

prediction variables, [6] reports that prediction accuracy is not heavily dependent on the size of pool. We have elected to use pools of 25 SNPs for test data sets with 200 total SNPs. We use $40,000$ trees to achieve convergence.

3.4.3 Advantages and Disadvantages

RFs have the potential to outperform the tree algorithms on which they are based while maintaining the ability to work on large data sets. Since some trees are forced to work with suboptimal views, results are less likely to be dominated by a few very important polymorphisms at the expense of less strong but important features. In addition, [4] claims that RF does not overfit data and uses very little memory. The importance score provided by the permutation test is a statistical measure of feature importance.

A disadvantage that is not shared with SVM and MDR is that the tree algorithm utilized by RF requires a marginal effect in at least one SNP if it is to identify interacting sets of SNPs [12]. Power to detect interactions with no main effect is expected to be reduced. To compensate, trees are also grown where pairs of SNPs form a prediction node. In effect, this creates a data set with $\begin{pmatrix} n \\ 2 \end{pmatrix}$ attributes. Results demonstrate that this method makes complexity quadratic in the number of SNPs and fails to differentiate between association and interaction, but that it does enable detection of pairs of SNPs that are in an interaction. Another implementation detail that deserves mention is that RF depends heavily on several parameters that must be adjusted for different problem sizes.

3.5 Data

Eleven genetic models are considered in this study, with each model designed to include a different alternate hypothesis. Genetic models were created by Carl Langefeld.

Single locus models were based on known genes discovered in autoimmune disorder studies [17]. Interactions were based on combinations of common models of Mendelian inheritance. Similar models were used in [23] for testing machine learning algorithms.

We include three models with a single ground truth SNP with a given penetrance and fixed background rate of disease. These models are designed to test the ability to detect a single locus association as well as to test the type I error of detecting an association or interaction between two SNPs when only one is known to be associated. Four models are included that contain a pair of SNPs that are associated with the phenotype. Two models have a pair of SNPs that are in a statistically interacting pair but that also possess marginal effects. Another two SNP model contains SNPs that are interacting but that have no marginal effects. The last model consists of two SNPs with independent main effects and is designed to test type I error of detecting an interaction when association but not interaction is present. Higher order models of size 3 and 5 are also considered. In each case, the SNPs in the model are interacting, but they also have marginal effects. All data sets contain a total of 200 SNPs and 2000 individuals split evenly among cases and controls. Of those 200 SNPs, between one and five are ground truth (GT) which means that case or control was assigned based partially on those SNPs. For each model, 1000 data sets were created using custom software written by the author and Joshua Grab.

Models considered in this thesis contain between one and five SNPs. We choose to focus on lower order models for both practical and biological reasons. Many algorithms considered have complexity $O(n^k)$ to detect k order subsets in n SNPs, making detecting of large models infeasible. Also, it is hypothesized that low order models are more common in actual genomes than higher order models. Higher order models can be formed when, for instance, several redundant copies of a gene all contain a deleterious mutation. If a single mutation is removed, the "correct" gene copy could

be utilized by the cell.

3.5.1 Single-SNP Dominant Model

We constructed data sets with a single SNP with MAF= 0.10 and the penetrance defined in Table 3.1. A sporadic rate of 0.40 was added to simulate the influence of unmeasured factors on phenotype. Unmeasured factors might include environmental effects or genetic factors that were not measured. Sporadic rates apply to all individuals, which means that 40% of individuals were randomly assigned to case regardless of their genotype.

Table 3.1: Single-SNP Dominant Penetrance (A: dominant allele.) For a given individual with a known genotype at the locus of interest, the probability of disease p is given by the table. Sporadic rates, which are added later and independently, are not included in this table. All tables in this section should be interpreted in the same manner.

Locus	AA	Aa	aa
p	0.0	0.5	0.5

3.5.2 Single-SNP Recessive Model

We constructed data sets with a single SNP with MAF= 0.40 and penetrance defined in Table 3.2. Cases were also assigned with a sporadic rate of 0.36.

Table 3.2: Single-SNP Recessive Penetrance (see Table 3.1)

Locus	AA	Aa	aa
p	0.0	0.0	0.5

3.5.3 Single-SNP Additive Model

We constructed data sets with a single SNP with MAF= 0.15 and penetrace defined in Table 3.3. This is an additive model created with a logistic model with

$\beta_0 = logit(0.05)$ and $\beta_1 = logit(3.0)$. The first coefficient can be interpreted as an environmental effect while the second is the odds ratio of the risk allele.

Table 3.3: Single-SNP Additive Penetrance (see Table 3.1)

Locus	AA	Aa	aa
p	0.05	0.13636	0.24752

3.5.4 Two-SNP Model with Penetrance in Major Allele

We constructed data sets with the MAFs = 0.25 and penetrance defined in Table 3.4. Each cell in the table describes the probability of disease (without considering sporadic rate) for an individual that has the genotype listed in the row and column. In addition, we included a sporadic rate of 0.05.

Table 3.4: Two SNP Major Allele Penetrance. Intrepretation is similar to Table 3.1 but utilizes two SNPs.

Locus	AA	Aa	aa
BB	0.60	0.60	0.00
Bb	0.60	0.60	0.00
bb	0.00	0.00	0.00

3.5.5 Two-SNP Model with Penetrance in Minor Allele

We constructed data sets with the MAFs = $0.20, 0.30$ and penetrance defined in Table 3.5. In addition, we included a sporadic rate of 0.10.

3.5.6 Two-SNP Model with Additive Penetrance

We constructed data sets to satisfy the model $\beta_0 + \beta_1 \text{SNP}_1 + \beta_2 \text{SNP}_2 + \beta_3 \text{SNP}_1 \text{SNP}_2$ with $\beta_1 = \beta_2 = 0, \beta_0 = logit(0.05)$, and $\beta_3 = logit(3.0)$. Penetrance is defined in Table 3.6. MAFs were both 0.20.

Table 3.5: Two SNP Minor Allele Penetrance (see Table 3.4)

Locus	AA	Aa	aa
BB	0.00	0.00	0.00
Bb	0.00	0.25	0.25
bb	0.00	0.25	0.50

Table 3.6: Two SNP Additive Penetrance (see Table 3.4)

Locus	AA	Aa	aa
BB	0.05	0.05	0.05
Bb	0.05	0.13636	0.32143
bb	0.05	0.32143	0.8100

3.5.7 Two-SNP Model with No Marginal Effect

We constructed data sets with MAFs = 0.20 and 0.30, respectively. Penetrance was designed to include only an interaction term and is presented in Table 3.7. No sporadic rate was included.

Table 3.7: Two SNP Interaction Only (see Table 3.4)

Locus	AA	Aa	aa
BB	0.0	0.0	0.20
Bb	0.0	0.20	0.0
bb	0.20	0.0	0.0

3.5.8 Two-SNP Model with No Interaction Effect

We constructed data sets with MAFs = 0.15 and 0.30, respectively. Penetrance was designed as the intersection of the dominant and recessive single SNP models. To construct these sets, we built two data sets with a proportion of cases equal to $1 - \sqrt{2}/2$. One set contained only the GT SNP while the other contained 199 SNPs including the GT SNP. Final sets were created by concatenating the two sets and

using a logical OR relationship to define case status. A sporadic rate of 0.35 was also included.

3.5.9 Three-SNP Model with Full Penetrance

We constructed data sets with MAFs $= 0.25, 0.20$, and 0.20. The model is recessive in the SNP with MAF $= 0.25$ and dominant in the other two SNPs with penetrance equal to 1.00.

3.5.10 Three-SNP Model with Partial Penetrance

We constructed data sets with MAFs $= 0.40, 0.25$, and $0, 25$ for SNPs A,B, and C. The model is ordinal in all three SNPs and has the following description:

$$\text{Prob(disease}|$$
$$\text{heterozygous for at least two loci and first genotype not } AA) = 0.50$$
$$\text{Prob(disease}|$$
$$\text{homozygous for minor allele at two two loci and first genotype not } AA) = 1.0$$

In addition, a sporadic rate of 0.10 was added to the data.

3.5.11 Five-SNP Model

We constructed data sets with 5 SNPs under the alternative hypothesis, all MAF $= 0.30$. The probability of disease is 1.00 if at least one copy of the minor allele is present at each locus and 0.00 otherwise.

3.6 Testing Philosophy and Methods

Tests of statistical power, defined here as the probability of rejecting a null hypothesis when the alternative is true, must be interpreted in light of the null hypothesis that is under consideration. For SVM, the null hypothesis for a given set of SNPs is that there is no association between the set of SNPs and the phenotype of interest. MDR's CV test and fitness landscape interpretations test the null hypothesis that the best model the algorithm discovers is unassociated with the phenotype of interest (for more details, see Section 3.6.1.) RF's test of joint importance has the null hypothesis that the pair of SNPs is not associated with the phenotype.

All three algorithms can use permutation tests to get an empirical p-value testing the validity of the null hypothesis, though this has only been attempted in the case of MDR and RF. In addition to the potential for a test of significance, all three algorithms rank SNPs or sets of SNPs of a given size, and the rank is used as a comparison between algorithms.

In all three cases, the null hypothesis should not be confused with the similar hypothesis that there is statistical epistasis between SNPs in a set. Statistical epistasis is defined as a non-additive relationship between a set of SNPs and a phenotype and is what we call interaction [11, 26, 36]. (Epistasis as it is defined here should not be interpreted as the original definition, which required that one gene suppresses another. See [11] for more information.) Using permutation testing as described, only the null hypothesis of no association between the set of SNPs and the phenotype can be tested. Particulars including the nature of the association (epistatic, additive, or marginal in one or more subsets) are not uncovered. A permutation test can test H_0 that no interaction exists if it permutes SNP values within cases and controls, thus preserving marginal effects.

For an n-way model, SVM, MDR, and RF return a fitness landscape that allows

the comparison of the test statistic (10-fold CV for RF, SVM and accuracy on test data for MDR) across all models of a given size. We test the rank of the test statistic for a ground truth SNP or set of SNPs among all tests.

ADTrees and Bagged ADTrees, like MDR, SVM, and RF, can not directly test for the presence of interaction. In fact, due to their need for a marginal interaction to "seed" a tree, we expect them to exhibit low power to detect pure interactions.

LR models are designed to test the null hypotheses described in Section 3.1. For instance, two-way test of interaction using logistic regression tests that an interaction is present ($\beta_3 \neq 0$). It should not detect a set of SNPs as interacting if only a subset is associated or if there are marginal effects with no interaction. The criteria that are used to reject H_0 for each algorithm are presented in Table 3.8.

For all models, we report power, type I error, and positive predictive value. We were unable to utilize sensitivity and specificity, which are common in the machine classification community because the results of our tests are features used for classification rather than classified instances directly. Specificity, which is defined as the percentage of time H_0 is not rejected when it is actually true, is always nearly 99%. We resorted to the statistics most commonly used by the feature detection or information retrieval community, which are more focused on true and false positives than the very large true negative pool [22].

3.6.1 Testing the Several Hypotheses in MDR

MDR is one of the more mature machine learning algorithms for computational genetics, and as such it is important to consider all of the competing methods of interpretation. In this section, we lay out all three methods and the rationale for each.

The fitness landscape test is modeled after the SVM test in that both methods operate solely on the reported test statistic of accuracy of classification for each model. We define H_0 as classification accuracy does not exceed that of a random SNP or set

Table 3.8: Criteria for Rejection of H_0.

Algorithm	H_0[a]	Rejection Criteria
LR[b]	$\beta_1 \neq 0$	$p < 0.05$
LR2[c]	$\beta_3 \neq 0$	$p < 0.05$
SW	No SNPs remain in final model.	Entry and exit at $p < 0.01$. H_0 rejected if model is non-empty.
ADTree[d]	SNP (or set) does not appear in the AD Tree.	SNP appears in tree containing 5% of the SNPs.
BADT1[e]	SNP (or set) does not appear in subpath in 4% bootstrapped trees.	SNP(s) appear in subpath in $>$ 4% of trees.
BADT2	SNP(s) form path in 4% bootstrapped trees.	SNP(s) form path in $>$ 4% of trees.
MDR (Fitness)	SNP (or pair) classification accuracy does not exceed that of random SNP or pair. (See Section 3.6.1.)	Classification accuracy in top 5%.
MDR (CV)	Best model is not consistent across CV folds. (See Section 3.6.1.)	$p < 0.05$ where distribution derived empirically.
MDR (Perm)	Classification using best model on out-of-bag individuals is not above average for best models (See Section 3.6.1.)	$p < 0.05$ where distribution derived from permutation test.
RF	Classification accuracy does not exceed that of random SNP or pair.	$p < 0.05$ under permutation test (permute phenotype.)
SVM	Classification accuracy does not exceed that of random SNP or pair.	Classification accuracy in top 5%.

[a] All null hypothesis except for SW are formulated for a specific SNP or set of SNPs.

[b] LR is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1$.

[c] LR2 is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1 + \beta_2 \text{SNP}_2 + \beta_3 \text{SNP}_1 \text{SNP}_2$ and $H_0 : \beta_3 = 0$.

[d] ADTrees were built using five percent of the data, so under the null hypothesis, there is a 5% chance a SNP will be included in the tree. There is a 0.05^2 probability that a pair will be included in the tree.

[e] We did not test multiple values of the threshold in this thesis. Note that a threshold of, say, 5% of trees does not correspond to a type I error of 5%.

of SNPs and rejection occurs if a model ranks in the top 5% in terms of classification accuracy. Since almost all models that are tested in an exhaustive test contain SNPs that are under H_0, the range of exhaustive tests does produce an estimate of the distribution of the test statistic under the null hypothesis. Both of the permutation tests return a p-value that should be used to judge H_0 that the best model is significant. Table 3.9 demonstrates the definition of power and type I error for these two tests.

Table 3.9: Power and type I error for best model returned by MDR. Type 2 error is marked for completeness but is not discussed.

		Model status	
	P-val range	GT	NGT
Maximum model:	$p > 0.05$	T2	—
	$p \leq 0.05$	Power	T1

For CV, we use a p-value calculated from the test data in each CV iteration. This is the most conservative test for MDR, as classification is over individuals that were not used in the production of the model and only the best model is even considered. Permutation testing utilizes a combinatorial permutation algorithm that preserves marginal effects to explicitly test for interaction. We consider the rank of the classification accuracy on out-of-bag individuals across 10 CV folds. Private communication with the software authors indicated that this test was the prefered method of interpretation.

3.6.2 Testing Environment

All tests except stepwise LR were performed on the DEAC Linux cluster, which schedules jobs on 152 Xeon E5430^{TM}8 CPU processors running at 2.66 Ghz. Each node has 16GB of RAM. The DEAC runs Redhat Enterprise Linux^{TM}with kernel version 2.6.9 and utilizes Maui^{TM}for scheduling on the cluster nodes. Algorithms were coded in FORTRAN77, C++, and Java and were controlled using Perl. Stepwise

Table 3.10: Table of algorithm abbreviations.

Algorithm	Abbreviation
LR	Logistic regression testing single SNP
LR2	Logistic regression testing interaction between two SNPs
SW	Stepwise logistic regression
ADTree	Alternating decision tree
BADT1	Bagged alternating decision tree using subpaths as features
BADT2	Bagged alternating decision tree using full paths as features
MDR (Fitness)	Multi-factor Dimensionality Reduction using fitness landscape.
MDR (CV)	Multi-factor Dimensionality Reduction using cross-validation test of model consistency
MDR (Perm)	Multi-factor Dimensionality Reduction using permutation test of model accuracy.
RF	Random forests
SVM	Support vector machines

logistic regression was run on a Sun^{TM}machine with 8 sparcv9^{TM}CPUs running at 900 Mhz. This machine has 16GB of RAM and is running Solaris 5.9^{TM}. The algorithm was programmed in SAS^{TM}.

Chapter 4: Results

4.1 Single-SNP models

The test data under development includes three models that include only a single ground truth SNP. For each model, we report the power of each algorithm to detect the SNP as well as type I error of rejecting several null hypotheses. Results are as follows. Table 4.2 contains type I error estimates for each model and algorithm under $H_0 : \beta_1 = 0$. All algorithms appear to have the correct type I error, except bagged ADTrees, MDR (CV), stepwise LR, and RF. See discussion for further explanation. Table 4.3 contains power estimates to detect a SNP under a dominant, additive, and recessive model for all algorithms. Using McNemar's test (equation 1.3) [1], we compared powers for each algorithm. Tables 4.4, 4.6, and 4.5 contain comparison results for the dominant, additive, and recessive models, respectively. With few exceptions, we were able to demonstrate statistically significant differences between algorithms.

In general, our tests do not allow us to present theoretical reasons why one algorithm might perform better than another on a specific models. Models differ in several parameters, especially MAF and sporadic rate. The lone exception is LR's performance on an additive model, which is as expected since LR is using an additive model.

Another form of type I error for algorithms that consider pairs of SNPs is the detection of an association when none is present. We first consider the case of two SNPs that satisfy $H_0 = \beta_1 = \beta_2 = \beta_3 = 0$, i.e. of two SNPs that have neither marginal effects nor an interaction, and results are presented in Table 4.7. Several algorithms have low type I error due to the stringent requirements for inclusion of

a SNP under H_0 in the final model. Testing type I error of detecting two SNPs that jointly associate when one is GT is also possible. We examine data satisfying $\beta_1 \neq 0, \beta_2 = \beta_3 = 0$ and test $H_0 : \beta_3 = 0$. Algorithms that detect group association rather than interaction generally produce elevated type I error of this type. Table 4.8 contains the type I error of detecting an association between two SNPs where one is GT. While most of the conservative tests in Table 4.7 remained so, RF, MDR, and SVM all demonstrated type I error consistent with their power in the same models.

4.2 Two SNP Models

Many genetic disorders are the result of two or more independent causes, and the first model with two SNPs is designed to test power and type I error in this situation. Table 4.9 contains power and type I error of detecting an interaction between two SNPs with marginal effects but no interaction. Data satisfies $\beta_1 \neq 0, \beta_2 \neq 0, \beta_3 = 0$ and the test is $H_0 : \beta_3 = 0$. Power is defined as detecting one (power of union) or both (power of intersection) SNPs using a pair of independent, single-SNP tests. Type 1 error is defined as detecting an association between the two SNPs. Once again, MDR (Fitness), SVM, and RF demonstrate elevated type I error. ADTrees also demonstrate very elevated type I error if the entire tree is used to judge interactions. See the discussion for further clarification of the tests that each interpretation of ADTrees represents. The other two ADTrees interpretations and BADT1 demonstrate type I error outside of the 95% confidence intervals.

Four models that contain two GT SNPs are designed to include an interaction term. Three models also exhibit marginal effects in both SNPs, while one exhibits a pure interaction. With the exception of stepwise logistic regression, the algorithms considered in this section work on each pair of SNPs in succession, and their power and type I error will be determined using the criteria described in Table 3.8. Power

is defined as rejecting $H_0 : \beta_3 = 0$ for each model. The exception is stepwise LR, which tests for multi-locus association rather than for interaction. When 2 SNPs are interacting without marginal effects, detection of SNPs by stepwise LR is type I error. The value was consistent with expectations and was omitted from the table to avoid confusion. Table 4.10 contains power results for all algorithms on 2 SNP models. Type I error is described in Table 4.7 and is not repeated. As the results varied between models, see the discussion for a summary.

4.3 Positive predictive value

Positive predictive value (PPV) is the probability that a rejected null hypothesis is actually false. We define PPV as the ratio of true positives to true positives plus false positives, so a PPV of 1 indicates that every time H_0 was rejected, it was actually false. Power and PPV, while related, reveal different characteristics of a test. Power gives the odds of detecting a model that is not under H_0 while PPV gives the probability that a model that is detected is not under H_0. For instance, RF under the test of two SNPs with penetrance on the major allele had power of 1.000 compared to 0.560 for BADT1. However, the aggregate type I error of detecting a two SNP association when one is not present is high, giving a PPV of 0.360 for RF. BADT1 has lower type I error, so PPV is 0.859. The correct interpretation of these results is that RF will detect more pairs of SNPs in joint association but that pairs detected by BADT1 are more likely to be under H_A.

PPV is presented for H_0 : no interaction. Aggregate T1 error for models that include two SNPs is computed by summing the three ways to reject H_0 for a pair of SNPs when there is not an interaction: neither SNP is marginally significant, one of the two SNPs is marginally significant and the other is not, and both SNPs are

marginally but independently significant. The final equation for T1 is give by

$$T1(2 - SNP) = p(\text{rej } H_0|\text{neither SNP GT}) +$$
$$p(\text{rej } H_0|\text{both SNPs GT, no int}) + 2p(\text{rej } H_0|\text{one SNP is GT, other not})$$

Tables 4.11 and 4.12 contain PPV for each algorithm for each model. Table 4.11 demonstrates that almost all algorithms have PPV above 0.900 on single-SNP models, which means that a SNP that is detected by any of these algorithms is ground truth with probability > 0.90. The lone exception, MDR (CV) demonstrates that PPV is ill posed if power and type I error are near zero. We did not compute PPV for MDR (Perm) because it will also be ill posed. Multiple possible configurations that are under the alternative hypothesis drive aggregate T1 error up and subsequently reduce PPV for all algorithms that operate on two SNPs.

Stepwise LR has the highest PPV for single-SNP models despite containing on average two non-ground truth SNPs in the final model. This results from the reduction of the set of SNPs potentially containing a ground truth SNP from $n = 200$ to $n = 3$. Bagged ADTrees gives high PPV for single-SNP models due to its robustness against noise.

Despite having lower power in several of the models, ADTrees rank among the highest PPV in all models except for the model with pure interaction. The average rank of ADTrees (same head) is 2.333 for the two SNP models that include interaction and marginal effects, higher than the average for any other model. Also, given the very low type I error of detecting a three SNP interacting pair for both ADTrees and bagged ADTrees, PPV would be near one for any model detected with size >= 3 SNPs.

Table 4.1: Power redefinition for detection of n SNP models, $n > 2$

ADT	Not model size specific. Test power to detect whole model and ability to detect 2 SNP subset.
BADT	Same as ADT.
SW	Same as ADT.
RF	Test power to detect 2 SNP subset.
MDR	Same as RF.
SVM	Same as RF.
LR2, Marginal LR	Same as RF.

4.4 Three and Five SNP Models

Exhaustive search over order 3 or higher subsets of a large data set proved infeasible for several algorithms. MDR (Fitness, CV) performed quickly enough to justify checking 3−way interactions, but RF, SVM, LR2, and Marginal LR were incapable of completing exhaustive searches in time. We chose not to pursue detection of order 3 subsets using MDR as power for CV and permutation and type I error for fitness did not justify the examination. For GWAS size studies with n SNPs, without extensive culling of data, and subsequent inability to detect interactions with no marginal effects, exhaustive search of size k subsets is $\Theta(n^k)$ and requires exceptional computational resources. As a result, we measure power in these three algorithms as the ability to detect smaller subsets of the SNPs for those algorithms that perform exhaustive search. Table 4.1 describes the interpretation changes required for each algorithm. When performing power to detect 2−SNP subsets, we report the power to detect all subsets, the power to detect a single subset, and the average number of subsets detected. Results are in Table 4.13. We also include power to detect whole models for ADTrees, which is designed to have low enough complexity to justify large data set searches.

4.5 Time Measurements

Time complexity is important as data sets grow towards GWAS sizes and time results in seconds are presented in Figure 4.1. Algorithms that rely on exhaustive testing or even heuristic testing to approximate exhaustion do not scale well even to a few thousand SNPs. Single-SNP analysis has been utilized effectively on as many as 1 million markers. Bagged ADT and ADT scale linearly with the number of SNPs but quadratically with the number of SNPs in the tree. If model size is kept constant, we anticipate lower type I error and power for larger data sets, so it is wise to scale model size. While this leads to $O(n^2)$ complexity, the constant is lower than other algorithms. Further investigation is warranted to understand the relationship between data size, model size, power, and type I error. Bagging scales with ADTrees, but it is anticipated that the same small tree can be used. Run times in bagged ADTrees also depend on the number of significant paths in the data, which would be expected to increase linearly with data size. While the number of features that actually associate with disease should not continue to increase with more SNPs, the total number of features detected including false positives could continue to increase. Reconsidering the choice of the parameter that defines a feature as in a significant number of trees can help control the increase in the total number of features returned by the algorithm. Overall, tree based methods present the clearest opportunity for expansion to GWAS size data sets.

Table 4.2: Single-SNP Type I Error [a] ($H_0 : \beta_1 = 0$)

	Dom model	Add model	Rec model
LR[b]	0.049	0.053	0.048
SW(1)[c]	0.006	0.012	0.013
SW(2)[d]	0.855(2.066)	0.858(2.145)	0.842(2.058)
ADT	0.044	0.039	0.039
BADT1	0.032	0.025	0.028
BADT2	0.012	0.011	0.009
MDR (Fitness)[e]	0.053	0.038	0.040
MDR (CV)[e]	0.000	0.000	0.000
RF	0.071	0.087	0.07
SVM	0.057	0.038	0.048

[a] All algorithms except Stepwise LR act on single SNPs. The 95% confidence interval for a T1 error of 0.05 is $(0.0365, 0.0635)$.

[b] LR is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1$.

[c] T1 error is calculated by choosing a random SNP and calculating the fraction of time that it appears in a model.

[d] T1 error is calculated as the percentage of the time that any non-GT SNP appears in the model. Number in parenthesis is average number of non-GT SNPs in model. Using 200 SNPs all under H_0, there was at least one SNP in the final model in 0.850 of the runs with an average of 2.0560 SNPs in each model.

[e] MDR T1 error is reported over test of single-SNP models. Fitness is reported as the fitness landscape test. CV uses the definitions presented in Section 3.6.1.

Table 4.3: Single-SNP Power

	Dom model[b]	Add model[c]	Rec model[d]
LR[e]	0.640	0.689	0.665
LR (Bonf. adj)[e]	0.097	0.093	0.124
SW	0.397	0.468	0.427
ADT	0.359	0.397	0.557
BADT1	0.488	0.534	0.742
BADT2	0.179	0.155	0.179
MDR (Fitness)[f]	0.481	0.530	0.756
MDR (CV)[f]	0.000	0.001	0.000
RF	0.539	0.524	0.721
RF (Bonf. adj.)	0.249	0.478	0.261
SVM	0.528	0.570	0.480

[a] All algorithms except Stepwise LR act on single SNPs.

[b] Model: Penetrance: $P(disease|\text{one copy of minor allele}) = 0.5$, independent sporadic rate = 0.4, and MAF = 0.10

[c] Model: Additive penetrance with $\beta_0 = 0.05, \beta_1 = \log(2.0)$. MAF = 0.15.

[d] Model: Penetrance: $P(disease|\text{two copies of minor allele}) = 0.5$, independent sporadic rate = 0.36, and MAF = 0.40

[e] LR is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1$. The second row is using a Bonferroni correction.

[f] MDR T1 error is reported over test of single-SNP models. Fitness is reported as the fitness landscape test. CV uses the definitions presented in Section 3.6.1.

Table 4.4: Pairwise comparison of test of power in Dominant Model (Algorithms ordered by power.)

	SVM	RF	BADT1	MDR	SW	ADT	BADT2
LR[a]	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
SVM		0.5430	0.0006	< 0.0001	< 0.0001	< 0.0001	< 0.0001
RF[b]			0.0210	0.0074	< 0.0001	< 0.0001	< 0.0001
BADT1				0.4882	< 0.0001	< 0.0001	< 0.0001
MDR					< 0.0001	< 0.0001	< 0.0001
SW						0.0060	< 0.0001
ADT							< 0.0001
BADT2							

[a] LR is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1$.

[b] RF uses 100 iterations.

[c] All algorithms except Stepwise LR act on single SNPs.

[d] Model: Penetrance = 0.5, Sporadic rate = 0.4, and MAF = 0.10

Table 4.5: Pairwise comparison of test of power in Additive Model (Algorithms ordered by power.)

	SVM	BADT1	MDR	RF	SW	ADT	BADT2
LR[a]	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
SVM		0.0332	< 0.0001	0.0387	< 0.0001	< 0.0001	< 0.0001
BADT1			0.5977	0.5202	< 0.0001	< 0.0001	< 0.0001
MDR				0.7815	0.0115	< 0.0001	< 0.0001
RF[b]					< 0.0001	< 0.0001	< 0.0001
SW						0.0039	< 0.0001
ADT							< 0.0001
BADT2							

[a] LR is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1$.

[b] RF uses 100 iterations.

[c] All algorithms except Stepwise LR act on single SNPs.

[d] Model: Additive penetrance with $\beta_0 = 0.05, \beta_1 = \log(2.0)$. MAF = 0.15.

Table 4.6: Pairwise comparison of test of power in Recessive Model. (Algorithms ordered by power.)

	RF	BADT1	LR	ADT	SVM	SW	BADT2
MDR	0.0752	0.1698	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
RF[a]		0.0839	0.0069	< 0.0001	< 0.0001	< 0.0001	< 0.0001
BADT1			< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
LR				0.0007	< 0.0001	< 0.0001	< 0.0001
ADT					0.0969	< 0.0001	< 0.0001
SVM						< 0.0001	0.0046
SW							< 0.0001
BADT2							

[a] LR is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1$.

[b] RF uses 100 iterations.

[c] All algorithms except Stepwise LR act on single SNPs.

[d] Model: Penetrance = 0.5, Sporadic rate = 0.36, and MAF = 0.40

Table 4.7: Type I Error $H_0 : \beta_3 = 0$. Data simulated using $\beta_1 = \beta_2 = \beta_3 = 0$

	Dom model[b]	Add model	Rec model
LR2[c]	0.049	0.043	0.052
ADT (Same rule)	0.002	0.000	0.000
ADT (Same head)	0.000	0.000	0.000
ADT (Same tree)	0.004	0.000	0.002
BADT1	0.003	0.004	0.003
BADT2	0.006	0.007	0.007
Marginal LR[d]	0.001	0.007	0.002
MDR (Fitness)[e]	0.050	0.042	0.055
MDR (CV) [e]	0.002	0.000	0.002
RF	0.090	0.099	0.091
SVM[f]	0.04	0.04	0.04

[a] All algorithms act on pairs of SNPs. SW omitted as T1 error is defined exactly as presented in Table 4.2. The 95% confidence interval for a T1 error of 0.05 is $(0.0365, 0.0635)$.

[b] Models are same as described in Table 4.3.

[c] LR2 is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1 + \beta_2 \text{SNP}_2 + \beta_3 \text{SNP}_1 \text{SNP}_2$ and $H_0 : \beta_3 = 0$.

[d] Marginal threshold is ($p_{thresh} = 0.20$)

[e] MDR T1 error is reported over test of single-SNP models. Fitness is reported as the fitness landscape test. CV and Perm use the definitions presented in Section 3.6.1.

[f] SVM uses the first 100 data sets rather than the full 1000, which makes the 95% confidence interval $(0.007, 0.093.)$ We were unable to run all 1000 data sets because the algorithm takes roughly as long as the rest of the algorithms combined.

Table 4.8: Type I Error $H_0 : \beta_3 = 0$ Data simulated using $\beta_1 \neq 0, \beta_2 = \beta_3 = 0$

	Dom model[a]	Add model	Rec model
LR2[b]	0.050	0.049	0.052
ADT (Same rule)	0.012	0.009	0.009
ADT (Same head)	0.002	0.009	0.015
ADT (Same tree)	0.019	0.017	0.020
BADT1	0.000	0.000	0.000
BADT2	0.004	0.002	0.007
Marginal LR[c]	0.008	0.006	0.009
MDR (Fitness)[d]	0.243	0.302	0.558
MDR (CV)[d]	0.003	0.000	0.000
RF	0.477	0.687	0.473
SVM[e]	0.26	0.31	0.38

[a] Models are same as described in Table 4.3. The 95% confidence interval for a T1 error of 0.05 is $(0.0365, 0.0635)$.

[b] LR2 is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1 + \beta_2 \text{SNP}_2 + \beta_3 \text{SNP}_1 \text{SNP}_2$ and $H_0 : \beta_3 = 0$.

[c] Marginal threshold is ($p_{thresh} = 0.20$)

[d] MDR T1 error is reported over test of single-SNP models. Fitness is reported as the fitness landscape test. CV and Perm use the definitions presented in Section 3.6.1.

[e] SVM uses 100 data sets rather than 1000, which makes the 95% confidence interval $(0.007, 0.093.)$

Table 4.9: Power and type I error for two SNPs with marginal effect, no interaction.

	Power ($\bigcup$)[a]	Power ($\bigcap$)[b]	T1 error ($\beta_3 = 0$)
LR[c]	0.931	0.500	0.063
Stepwise LR[d]	0.797	0.294	——-
ADT (same rule)	0.813	0.280	0.090
ADT (same head)	0.813	0.280	0.090
ADT (same tree)	0.813	0.280	0.280
BADT1	0.904	0.411	0.089
BADT2	0.547	0.147	0.064
Marginal LR[e]	0.931	0.500	0.042
MDR (Fitness)	0.968	0.555	0.917
MDR (CV)	0.003[f]	0.000	0.003
MDR (Perm)	———	———	0.000
RF[g]	0.80	0.13	0.76
SVM[g]	0.918	0.628	0.16

[a] Power over union is defined as rejecting the null hypothesis for either SNP that is under H_A.

[b] Power over intersection is defined as rejecting the null hypothesis for both SNPs that are under H_A.

[c] LR uses model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1$ for power columns and $logit(y) = \beta_0 + \beta_1 \text{SNP}_1 + \beta_2 \text{SNP}_2 + \beta_3 \text{SNP}_1 \text{SNP}_2$ and $H_0 : \beta_3 = 0$ to test T1 error of detecting an interaction.

[d] Stepwise LR should detect both SNPs in the model, which is measured here as power ($\bigcap$). Falsely detecting an interaction is impossible because interactions are not detected.

[e] Marginal threshold is ($p_{thresh} = 0.20$). Marginal LR for power is the standard LR test.

[f] MDR (CV) returns only a single model, power to detect two models in the same data set is 0 unless an iterative scheme is employed.

[g] RF and 2-SNP SVM results use 100 iterations.

[h] Model: The first SNP is a dominant model, MAF = 0.15. Second SNP is a recessive model, MAF = 0.30. A sporadic rate of 0.35 was also included. Marginal effects are independent.

Table 4.10: Power to reject $H_0 : \beta_3 = 0$ when two SNPs have marginal effects and/or interaction.

	Major Allele Pen.[a]	Minor Allele Pen.[b]	Add model[c]	Only Interaction[d]
LR2[e]	0.155	0.459	0.496	0.096
ADT (same rule)	0.535	0.061	0.020	0.046
ADT (same head)	0.535	0.061	0.020	0.046
ADT (same tree)	1.000	0.071	0.023	0.055
BADT1	0.560	0.020	0.004	0.013
BADT2	0.076	0.022	0.004	0.008
Marginal LR[f]	0.155	0.167	0.012	0.035
MDR (Fitness)[g]	1.000	0.572	0.189	0.841
MDR (CV)[g]	1.000	0.000	0.000	0.005
MDR (Perm)[g]	0.000	0.000	0.000	0.000
RF[h]	1.000	0.45	0.31	0.41
SVM[h]	1.000	0.39	0.17	0.59
SW[i]	0.999(0.999)[f]	0.008(0.023)	0.010(0.162)	——

[a] Model: Dominant in both major alleles, penetrance = 0.6, sporadic rate of 0.05. Both minor allele freqs are 0.25.

[b] Model: Penetrance = 0.25 if possess at least one copy of minor allele at each locus and = 0.50 if 2 copies at each locus. Sporadic rate of 0.10. MAFs are 0.20 and 0.30.

[c] Model: Additive with $\beta_0 = 0.05, \beta_3 = \log(3.0)$. MAFs are 0.20.

[d] Model: Penetrance for AAbb, AaBb, aaBB is 0.20. MAFs are 0.20 and 0.30. No sporadic rate.

[e] LR2 is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1 + \beta_2 \text{SNP}_2 + \beta_3 \text{SNP}_1\text{SNP}_2$ and $H_0 : \beta_3 = 0$.

[f] Marginal threshold is ($p_{thresh} = 0.20$)

[g] See Section 3.6.1 for further information on the varieties of MDR tests.

[h] RF and SVM Results use 100 iterations.

[i] SW is testing for presence of both GT SNPs but not for interaction, which is a different H_0. Power is presented as power to detect both SNPs followed by power to detect either SNP in parentheses.

Table 4.11: PPV for Single-SNP Models

	Dom model[a]	Add model	Rec model
LR[b]	0.928	0.929	0.933
SW	0.985	0.975	0.970
ADT	0.891	0.911	0.935
BADT1	0.938	0.955	0.964
BADT2	0.937	0.934	0.952
MDR (Fitness)[c]	0.900	0.933	0.950
MDR (CV)[c]	0.000	1	0.000
RF	0.883	0.857	0.905
SVM	0.903	0.938	0.909

[a] Model described in Table 4.3.
[b] LR is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1$.
[c] MDR T1 error is reported over test of single-SNP models. Fitness is reported as the fitness landscape test. CV and Perm test use the definitions presented in Section 3.6.1.

Table 4.12: PPV for two SNP models

	Major Allele Pen.[a]	Minor Allele Pen.	Add model	Only Interaction
LR2[b]	0.424	0.685	0.702	0.312
ADT (same rule)	0.828	0.355	0.153	0.292
ADT (same head)	0.828	0.355	0.172	0.292
ADT (same tree)	0.760	0.183	0.068	0.148
BADT1	0.859	0.179	0.042	0.008
BADT2	0.490	0.218	0.048	0.092
Marginal LR[c]	0.944	0.739	0.169	0.372
MDR (Fitness)[d]	0.370	0.252	0.100	0.331
RF[e]	0.360	0.180	0.148	0.187
SVM[e]	0.536	0.311	0.164	0.406

[a] Models defined in Table 4.10.
[b] LR2 is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1 + \beta_2 \text{SNP}_2 + \beta_3 \text{SNP}_1 \text{SNP}_2$ and $H_0 : \beta_3 = 0$.
[c] Marginal threshold is ($p_{thresh} = 0.20$)
[d] See Section 3.6.1 for further information on the varieties of MDR tests. MDR (CV) was removed due to low power.
[e] RF and SVM Results use 100 iterations.
[f] SW is testing for presence of both GT SNPs but not for interaction, which is a different H_0.

Table 4.13: Power to reject $H_0 : \beta_3 = 0$ for 2 SNP subsets when > 2 SNPs have marginal effects and interaction.

	Five SNP[b]	Three SNP A[c]	Three SNP B[d]
LR2[e]	0.931(0.000)	0.540(0.007)	0.997(0.429)
ADT (same rule)	0.204(0.103)	0.998(0.351)	0.846(0.778)
ADT (same head)	0.204(0.103)	0.998(0.351)	0.846(0.778)
ADT (same tree)	0.243(0.104)	0.998(0.984)	0.851(0.790)
BADT1	0.078(0.000)	1.000(0.791)	0.801(0.585)
BADT2	0.066(0.000)	0.994(0.000)	0.493(0.033)
Marginal LR[f]	0.516(0.000)	0.133(0.000)	0.976(0.328)
MDR	0.995(0.029)	0.658(0.000)	1.000(0.931)
RF[g]	0.89(0.04)	1.00(0.42)	1.00(0.94)
SVM[g]	0.98(0.00)	0.61(0.02)	0.99(0.74)

[a] Power is displayed as power to detect at least one subset and, in parenthesis, power to detect all subsets.
[b] Model: MAFs = 0.20. Penetrance = 1.0 if minor allele present at all loci.
[c] Model: MAFs $= 0.25, 0.20, 0.20$. Fully penetrant, recessive in first SNP, dominant in other two.
[d] Model: MAFS $= 0.40, 0.25, 0.25$. Ordinal in all three SNPs.
[e] LR2 is using model $logit(y) = \beta_0 + \beta_1 \text{SNP}_1 + \beta_2 \text{SNP}_2 + \beta_3 \text{SNP}_1 \text{SNP}_2$ and $H_0 : \beta_3 = 0$.
[f] Marginal threshold is ($p_{thresh} = 0.20$)
[g] RF and SVM Results use 100 iterations.

Table 4.14: Power to reject H_0 of no interaction (SW: association) 3 or 5 SNP models.

	Five SNP[b]	Three SNP A[b]	Three SNP B[b]
SW[c]	$0.000(0.018, 0.126)$	$0.001(0.031, 0.767)$	$0.167(0.645, 0.941)$
ADT (same rule)	0.103	0.351	0.778
ADT (same head)	0.103	0.351	0.778
ADT (same tree)	0.104	0.984	0.790
BADT1	0.001	0.789	0.584
BADT2	0.018	0.782	0.423

[a] Power is displayed as power to detect model with all GT SNP, except for SW.
[b] Models as in 4.13.
[c] Power is presented as power to detect whole model followed by power to detect two and then one GT SNP.

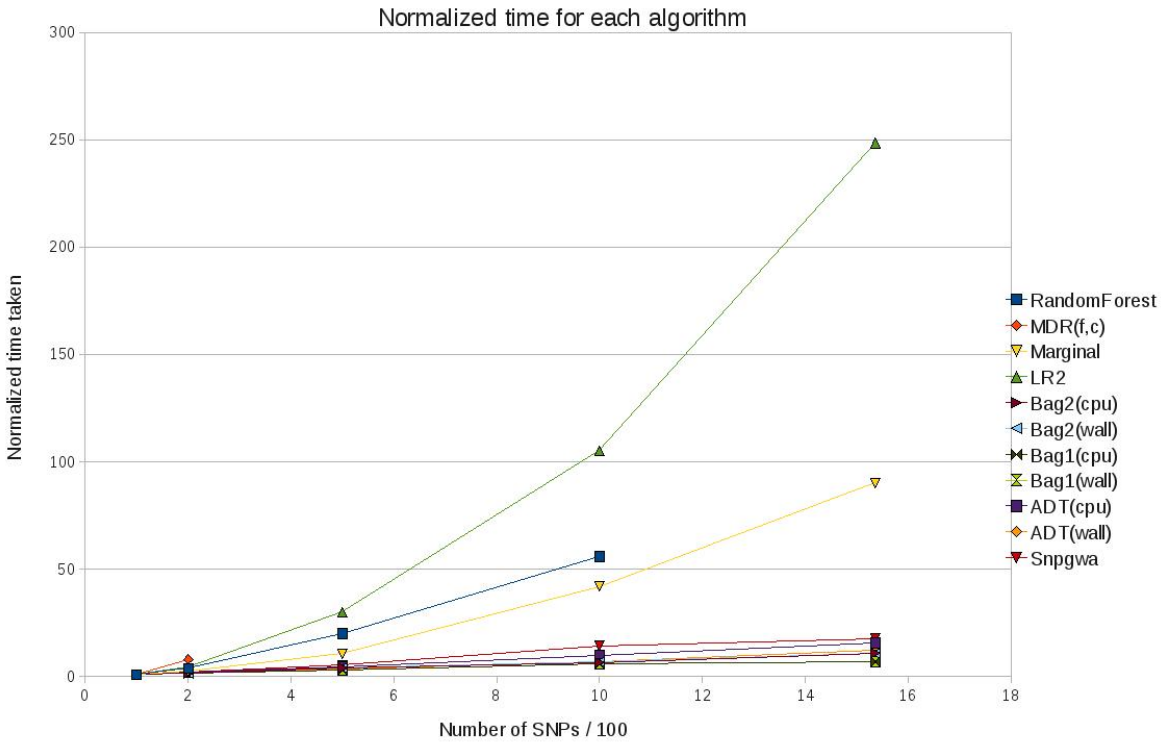

Figure 4.1: Run time relative to problem size averaged over ten runs per algorithm. Each algorithm's time was normalized so that the first point (100 SNPs) took one time unit. Note that only four algorithms in the chart have $O(n^2)$ complexity: MDR, LR2, Marginal LR2, and RF. Both RF and MDR have truncated curves due to memory limitations. Each process was alloted 4GB RAM. SVM is not shown, but was found to require quadratically many SVM solves. Data was created with between 100 and 1536 SNPs, all created under null hypothesis of no association and no interaction. ADT, BADT, and LR on single SNPs are all $O(n)$ algorithms. Note that we include wall time and parallel time for several algorithms. Using 4 CPUs, we saw 99% efficiency for BADT and ADT.

Chapter 5: Discussion

Results vary widely for different algorithms, models, and model sizes and underline the importance of correctly formulating hypotheses. For single-SNP models, only LR and bagged ADTrees over subpaths have power in the top 50% of all algorithms for each model. Logistic regression demonstrates the best performance overall, despite testing an additive model for penetrance functions that do not fit the additive model. Using model-specific logistic regression tests, we anticipated that LR would be the best algorithm in all three models, though Bonferroni correction reduces the power to 0.10. Since ADTrees and bagged ADTrees do not perform multiple statistical tests, they do not require a Bonferroni type correction.

Type 1 error varies widely. It was anticipated that type I error would be roughly 0.05 for most algorithms. For type I error of rejecting $H_0 : \beta_1 = 0$, bagged ADTrees (both types), stepwise LR, and MDR (CV) have abnormal results. Since a random SNP must be present in at least two trees to be counted in a subpath in these results, it makes sense that we would see type I error slightly more than $0.05^2 = 0.0025$ for bagged ADTrees over subpaths and even lower for bagging over full paths as the criteria for acceptance are even stricter. Stepwise LR demonstrates low type I error if calculated as the percentage of time a given random SNP appears in the data but is highly elevated if we only consider the presence or absence of a SNP in the model. Due to the composite null hypothesis that is tested by H_0, it is difficult to interpret the relationship between the entrance and exit criteria ($p < 0.01$) and the results presented here. Low power compared to other methods and a difficult interpretation lead us to recommend caution when using stepwise LR for detection of single-SNP

models, though any SNP in a SW model merits further investigation.

MDR's performance depends on the interpretation given to the results. Using the CV hypothesis test given in the results, it is hard to calculate either power or type I error. Several factors are at play. By limiting results to only the very top model discovered in the data, the test has low power to detect moderate effects in a noisy environment. This corroborates results in [8]. Even more striking was the absence of p-values in the bottom 5% of the theoretical distribution. For data constructed to satisfy H_0, p-values are uniformly distributed between 0 and 1, and we expect 50 of the 1000 tests in each model to have a p-value < 0.05. Only one or two values were that low per model. Figure 5.1 shows the distribution of p-values for data that satisfies H_0. P-values for the permutation test have the correct uniform distribution, and the figure was omitted. If limited to single-SNP models, the fitness landscape provides a test with correct type I error and relatively high power (the highest in one model) but the relative merits of this interpretation are lost by the inability to distinguish between interaction and association in a subset. We did not test MDR with a permutation test of significance for single-SNP models as it is designed to detect interactions.

Type I error of detecting an interaction between two SNPs when neither is ground truth was as expected in all algorithms, but several algorithms have elevated type I error of detecting an interaction between a pair of SNPs if one of the two is ground truth. Algorithms that test $H_0 : \beta_3 = 0$ (LR2, Marginal LR) have correct type I error. Algorithms that test a null hypothesis of no association or a composite null hypothesis of no interaction or association have elevated type I error. These algorithms are unable to differentiate between a set of SNPs that jointly associate with phenotype and a set of SNPs with a strict subset that associates with phenotype. The same pattern occurs in the test of type I error of detecting an association between two

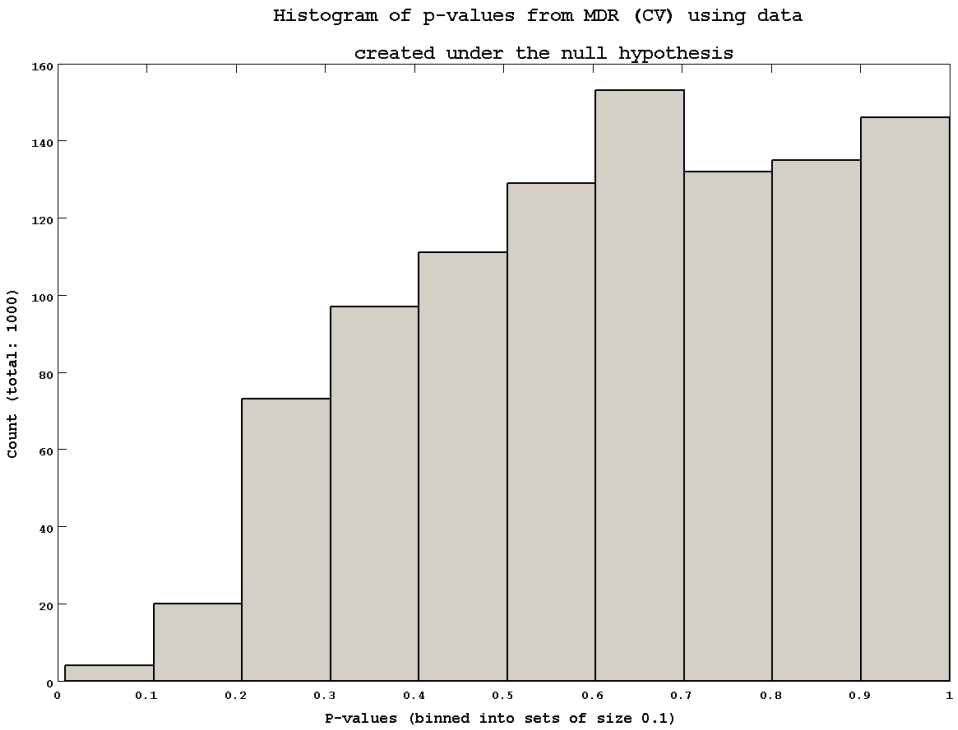

Figure 5.1: MDR (CV) has a nonuniform distribution of p-values on data that satisifies the null hypothesis. Data was constructed that had no association with the phenotype. We ran MDR and recorded the p-value from cross-validation test.

SNPs that have marginal but independent association. MDR (Fitness) and RF have type I error rates comparable to their respective power to detect a pair of SNPs with interaction and marginal effects because, as suggested previously, they are incapable of distinguishing between the null hypothesis that $\beta_1 = \beta_2 = \beta_3 = 0$ and the null hypothesis that $\beta_3 = 0$. SVM has elevated type I error but not to the same extent as seen in MDR (Fitness) or RF. By contrast, ADTrees has the lowest type I error rate of all algorithms that do not directly test $H_0 : \beta_3 = 0$. Interpreting ADT using the same head node or same path rules allowed the algorithm to differentiate marginal independent effects by placing them in separate parts of the tree that are interpreted in parallel. Interpreting ADT using the coarser rule of same tree does not distinguish independent marginal effects from interaction and should be treated as a test of the composite hypothesis of no association.

The models chosen to test power to detect two SNPs that exhibit interaction and marginal effects were in general more difficult to detect than the single-SNP models, with some algorithms failing to detect models at a nonrandom rate. ADTrees and bagged ADTrees in particular fail to detect three of the four models with greater than 10% power. An important factor influencing power is the model that an algorithm is designed to detect. For instance, the lack of a marginal effect makes it understandable that stepwise LR, ADTrees and bagged ADTrees would perform poorly when only an interaction is present or when marginal effects are weak. MDR (Fitness), SVM, and RF detect a pure interaction with nearly the same power as their type I error of detecting two SNPs with marginal effects but no interaction. The model with penetrance on the major allele is an example of the failure of logistic regression to detect a model due to an incorrect hypothesis about the form of the model. The model in question includes an epistatic effect from having two copies of the major allele in both SNPs. This might also explain why ADTrees places the two penetrating SNPs in different subtrees 46.5% of the time despite including both in the tree in 1000

out of 1000 iterations.

MDR (Perm) failed to show nonzero power or type I error for two locus models. Many of the problems with MDR (CV) mentioned earlier in this section also apply to MDR (Perm). We note, however, that the problem of incorrectly distributed p-values does not occur using the permutation test.

Models with three and five SNPs pose special problems for exhaustive search techniques, as computational complexity makes exhaustive search infeasible. LR2, MDR, SVM, and RF demonstrated power greater than 0.50 to detect pairs of interacting SNPs in all models, suggesting that in some cases a progression of $2, 3, \ldots$ up to n way models with culling of nonsignificant smaller interactions might be effective. One problem with ADT and BADT is the lack of a marginal effect suitable for starting the construction of a tree. ADT and BADT perform well on both three SNP models in both the test of two SNP subsets and the test for whole models, suggesting that more complicated models that include marginal effects should play to the strength of the ADTrees algorithm. The five SNP interaction includes a slight marginal effect, but it was not strong enough to provide a seed on which a tree could be constructed.

5.1 Conclusion

While the ADTrees algorithm has promise as a scalable machine learning algorithm for GWAS size studies, its power and PPV are highly dependent on the existence of marginal effects in at least one SNP involved in an interaction. RF has the same problem, which it overcomes by building trees in which every node is a pair of SNPs. We did not investigate this approach for ADTrees, though we suspect that it would suffer from an inability to distinguish association and interaction.

When ADTrees have adequate power, multiple interpretation methods provide for multiple tests. For instance, interaction can be tested by examining only subpaths or

subtrees. Association with the entire set or multiple independent subsets is possible by examining the entire tree as a unit. Bagging ADTrees appears to reduce type I error and to raise power to detect small models. Degredation in performance for larger models is a function of the more stringent requirements necessary to capture a longer path, and further investigation of corrective measures for this discrepancy in power is warranted. Positive predictive power measurements demonstrate that bagging decreases PPV in the absence of marginal effects (i.e. when we expect ADTrees to fail) while marginally boosting PPV when there is a marginal effect and we expect ADTrees to have good performance. Unlike the similar RF, bagging uses algorithms that can distinguish interaction and marginal association in a subset. Inclusion of bagging with other machine learning algorithms is an ongoing investigation.

All algorithms demonstrate the importance of understanding the null hypothesis that is tested by an algorithm and its interpretation. In particular, seemingly high power as seen in SVM, MDR (Fitness) and RF might lead to the acceptance of associated SNPs without considering the probability, given by PPV, that they are false positives. In particular, machine learning algorithms for detecting pairs of interacting SNPs must be certain not to conflate interaction with association, and all algorithms must take into account the limitations in the forms of models they are designed to detect. Intelligent permutation techniques provide one method of investigating H_0 : no interaction, but their complexity for high order methods combined with the inherent difficulties in high order exhaustive testing suggest that methods should be developed that directly penalize association without interaction. In general, there is a tradeoff between exhautive testing and reliance on marginal effects. Assuming the presence of marginal effects allows filtering of lower order subsets in order to reduce complexity and multiple testing problems, but the marginal effects assumption removes the ability to detect models that exhibit pure interaction over large sets of SNPs.

The widespread differences between algorithms and models suggests an ensemble approach to GWAS studies with machine learning techniques. Ensemble techniques utilize several different algorithms on the same data set. Reporting features that are detected by more than one algorithm has promise to increasing power, because different algorithms and their different null hypotheses cast a wider net. However, as new machine learning algorithms are developed and deployed, care must be taken to understand the specific strengths and weaknesses of each algorithm, and to avoid portraying any one algorithm as ideal for every model or situation. Machine learning has the ability to remove the dependence on specific genetic models as seen in regression techniques, and it may have some answers to the complexity challenges behind exhaustive, multi-allelic searches, but exuberance must not be lead to confusion about what kind of patterns a particular algorithm is able to detect versus those that it finds indistinguishable.

Chapter 6: SNPDoc: Integrating genomic data with statistical results

This chapter has been submitted as is to Bioinformatics (OUP) for publication. Formatting and references have been left unchanged and do not correspond to the rest of the document.

Authors: Richard T. Guy[1], Wei Wang, Miranda C. Marion[1], Paula S. Ramos[1], Ken Wilson[1], Timothy D. Howard[2], Carl D. Langefeld[1]

[1]Center for Public Health Genomics and Department of Biostatistical Sciences, Division of Public Health Sciences, Wake Forest University Health Sciences, [2]Center for Genomics and Personalized Medicine Research

Corresponding Author: Carl D. Langefeld, Ph.D.

Abstract

Summary: The integration of genomic data with the results of high-throughput statistical analyses is a critical need in this era of genome-wide association and eQTL analyses. SNPDoc (SNP Documenter) searches nine public databases for information on user-provided (potentially lengthy) lists of single nucleotide polymorphisms (SNPs) or genomic positions/regions and merges these results with user provided statistical results and other information. The program outputs html files that can be opened by readily available programs (e.g., MS-Excel) for easy manipulation.

Availability and Implementation: SNPDoc is available as a web service at https://wakegene.phs.wfubmc.edu. Source code is freely available at www.phs.wfubmc.edu/public/bios/gene/downloads.cfm.

Contact: clangefe@wfubmc.edu; tguy@wfubmc.edu

1. INTRODUCTION The advent of cost-effective, high-density genotyping of genetic markers with commercial arrays has lead to a proliferation of large-scale association studies. Efficient mining of these results for the most important associations is improved by integrating genomic information with statistical results. For example, prioritizing SNPs for replication based on genome-wide association studies often involves the selection of a subset of polymorphisms with comparable statistical evidence of association in the discovery cohort. Knowledge that a particular polymorphism resides within a region of higher a priori interest (e.g., promoter region, within or in the shore of a CpG island, interspecies conserved region, copy number variant), has a greater likelihood of altering gene protein function, or has a previously published association of interest can be very valuable. Because of the complexity of the genome, researchers must navigate through several online sources to obtain this information. Here we present a tool, SNP Documenter (SNPDoc), that integrates genetic information from multiple public websites with statistical or other information from an investigator to facilitate interpretation of results from large-scale genomic studies.

2. METHODS

SNP input consists of known SNPs (with rs numbers) and optional user-provided study results or other information to be displayed in the final output. Regional input consists of a chromosome and start and end positions. (For regional input, all known SNPs in the region are recorded and stored for processing as SNP input.) Positional input consists of chromosome and single positions for which a SNP name may not be known. SNPDoc executes informatics searches for the data detailed in Figure 1.

Core processing consists of queries to NCBI (Sherry et al., 2001; http://eutils.ncbi.nlm.nih.gov/entrez/query/static/eutils_help.html), Ensembl (Flicek 2010; http://uswest.ensembl.org/info/data/api.html), and UCSC (Kent 2002; Karolchik

2004; http://www.genome.ucsc.edu/) databases. NCBIs SNP database provides position, chromosome, and alias information about SNPs. SNPDoc also uses the NCBI SNP database to determine if the query SNP is located within a gene. Gene symbol, description, and aliases for any gene(s) containing the reference SNP are recorded from the NCBI Gene database. For positional input, the UCSC database is used to determine whether the position is located within a gene. Gene symbol, description, and aliases for any gene(s) containing the reference SNP are recorded from the NCBI Gene database. For SNPs or positions that are not in genes, we query the UCSC database for the nearest 5 and 3 genes within 500kbp. SNPs that are in genes are assigned an estimated risk score based on a minor modification of the algorithm found in FastSNP (Yuan 2006). Specifically, we return only the highest risk score from the set of functions returned by Ensembl. Finally, all SNPs and positions are queried against local tables of CpG islands (Bock et al., 2007) and structural variation regions (Daryl et al., 2007). The distance (bp) to the nearest CpG island (and whether the SNP alters a CpG site) and type of structural variation, if relevant, are recorded.

SNPdoc reports results as an html table suitable for web or spreadsheet access (Fig. 2). In addition, genomic information (but not statistical results or any other user-specific information) is stored in a local database for fast retrieval of repeat queries. Public access to the database is available at
https://wakegen.phs.wfubmc.edu/public/snpdoc/index.cfm and is currently designed to accept SNP input only.

3. IMPLEMENTATION

SNPDoc was implemented as a Perl script with web interface provided by ColdFusion. The script is designed to run on Linux, Windows, and Unix operating systems provided that Perl and several Perl modules described below have been installed. Web access to the online resources described in the Methods section is facilitated by the

LWP (Library for WWW in Perl) module, by the DBI (Database Interface) module, and by the Ensembl Perl API. For details on Perl and the specific modules utilized, see the Perl repository at CPAN (www.cpan.org.)

Web access is facilitated by a ColdFusion front end and a SQL Server database. Requests through the web server allow reuse of previously cached results. SNPs that are not in the database are added to a queue of outstanding requests that are captured every 5 minutes and added to the database. Upon completion, results are returned in the same order as submitted. Results are retrieved using a user ID and are available in both HTML format and Excel format.

4. RESULTS

Systemic lupus erythematosus (SLE) is a common systemic autoimmune disease with complex etiology and strong genetic influences on risk. The International Consortium on the Genetics of Systemic Lupus Erythematosus (SLEGEN, www.slegen.org) completed a genome-wide association study in women of European ancestry with a discovery cohort of 720 women diagnosed with SLE and 2,337 healthy women controls, and replication cohorts of 1,846 SLE affected women and 1,825 healthy women controls (SLEGEN et al. 2008). Here, we apply SNPDoc to the results in Table 1 of that paper. Specifically, the input file contained the SNPs with the strongest statistical association in the discovery and replication cohorts, their reference alleles, p-values from the tests of association, and the odds ratios and corresponding 95% confidence intervals. SNPDoc integrated these results with the genomic data. The results of the SNPDoc application allow the researchers to immediately observe that the most significant SNPs on chromosome 6 were within the well-known HLA region, and two other SNPs were within a relevant candidate gene, ITGAM. Interestingly, several of the SNPs are in proximity to known copy number variation or within a (CpG) island. The original application of SNPDoc to these data summarized the top

10,000 SNPs and aided in the selection of SNPs for the replication cohort reported in this study.

5. CONCLUSION

In this report we present a tool, SNPDoc for integrating genetic information from several online databases with user-specified information. SNPDoc facilitates wide area inspection of a variety of genomic data and aids in the selection of high-priority polymorphisms for replication studies. Access is available through a Perl script or via web access that allows for caching of popular requests.

REFERENCES Bock C., et al., (2007) CpG island mapping by epigenome prediction. PLoS Computational Biology 3(6):110.

Daryl J. et al., (2007) Variation resources at UC Santa Cruz. Nucleic Acids Research. 35:716-720,

Flicek P, et al., (2010) Ensembl's 10th year. Nucleic Acids Research 38:557-562.

Karolchik D, et al., (2004) The UCSC Table Browser data retrieval tool. Nucleic Acids Research 32:493-496.

Kent WJ, et al., (2002) The human genome browser at UCSC. Genome Research 12(6):996-1006.

Sherry ST, et al., (2001) dbSNP: the NCBI database of genetic variation. Nucleic Acids Research 29(1):308-11.

SLEGEN, et al., (2008) Genome-wide association scan in women with systemic lupus erythematosus identifies susceptibility variants in ITGAM, PXK, KIAA1542 and other loci. Nature Genetics 40, 204 210.

Yuan HY, et al., (2006) FASTSNP: an always up-to-date and extendable service for SNP function analysis and prioritization. Nucleic Acids Research 34:635-641.

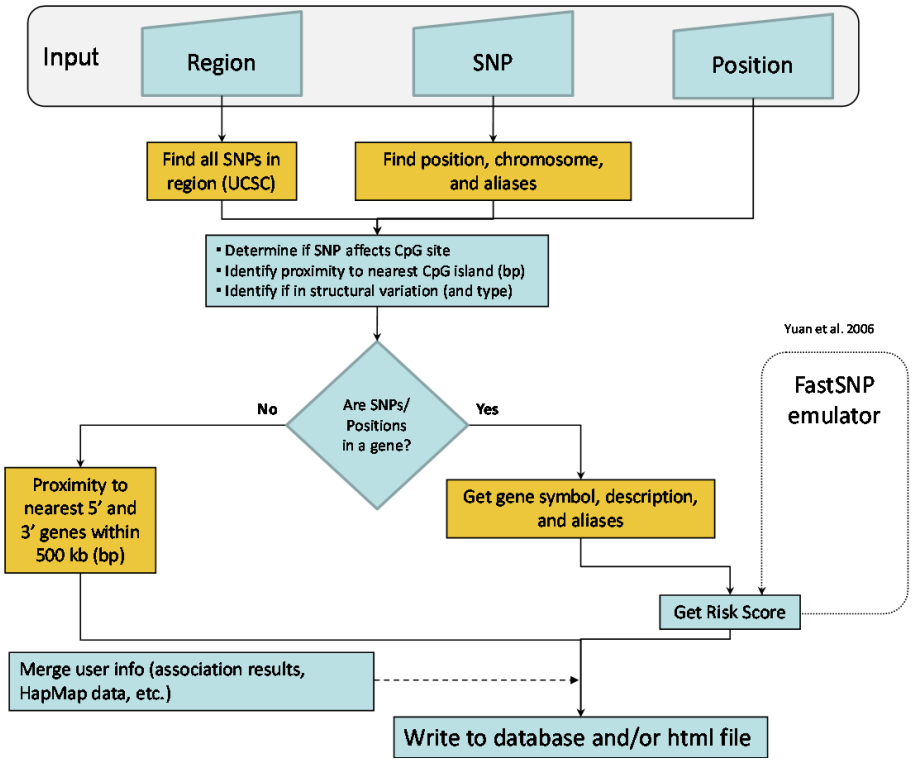

Figure 6.1: Figure 1. The Basic SNPdoc program flow. SNPdoc accepts regional, SNP, or positional input and produces an html output containing genomic and user submitted information.

MARKER	NCBI	UCSC	CHR	POSITION	NEAR GENE1	NEAR GENE2	GENENAME	DESCRIPTION	RISK	FUNCTION	CpGISLAND	VARIATION **	Reference Allele	P-Value	OR	95% CI
rs3131379	NCBI Link	UCSC Link	6	31829012			MSH5	mutS homolog 5 (E. coli)	0	Intronic With No Known Function	12698		A	1.71E-52	2.36	2.11-2.64
rs1270942	NCBI Link	UCSC Link	6	32026839			CFB	complement factor B	0	Downstream With No Known Function	7628 *	CN	G	1.27E-51	2.35	2.10-2.63
rs9888739	NCBI Link	UCSC Link	16	31220754			ITGAM	integrin, alpha M (complement component 3 receptor 3 subunit)	0	Intronic With No Known Function	29198 *		T	1.61E-23	1.62	1.47-1.78
rs1143678	NCBI Link	UCSC Link	16	31250506		94.514 kb from ITGAX	ITGAM	integrin, alpha M (complement component 3 receptor 3 subunit)	4	Mis-Sense (Splicing Regulation, Protein Domain Abolished)	-1		T	8.50E-14	1.4	1.28-1.53
rs4548893	NCBI Link	UCSC Link	16	31271994		32.823 kb from TRIM72			0	N/A	18432 *		A	2.36E-12	1.34	1.24-1.46
rs729302	NCBI Link	UCSC Link	7	128356196		23.149 kb from CALU	FAM71F2		0	N/A	8857	CN	C	2.00E-10	0.78	0.72-0.84
rs10279821	NCBI Link	UCSC Link	7	128470783			TNPO3	transportin 3	3	Intronic Enhancer	10965 *		T	6.50E-09	0.8	0.72-0.84
rs12537284	NCBI Link	UCSC Link	7	128505142		11.778 kb from KCP			0	N/A	22550		A	3.61E-19	1.54	1.40-1.70
rs4963128	NCBI Link	UCSC Link	11	579564			KIAA1542	PHD and ring finger domains 1	0	Intronic With No Known Function	238	CN	T	3.00E-10	0.78	0.73-0.85
rs6445975	NCBI Link	UCSC Link	3	58345217		133.242 kb from TNFSF4	PXK	PX domain containing serine/threonine kinase	0	Intronic With No Known Function	48998		C	7.10E-09	1.25	1.16-1.35
rs10798269	NCBI Link	UCSC Link	1	171576336		136.772 kb from PRDX6			0	N/A	136264	CN	T	1.11E-07	0.82	0.76-0.88

Figure 6.2: Table 1. SNPDoc output for association with Systemic Lupus Erythematosus (Harley 2008). Risk Score from the modified FASTSNP algorithm (Yuan et al., 2006) CpGIsland The number of basepairs from a known CpG island; * denotes that variation changes C or G in the CG pair, -1 denotes that the SNP is in a CpG island **Variation Denotes copy number (CN) and insertion/deletion (IN/DEL)

Glossary

Admixture Genetic admixture occurs when previously homogenous genetic populations interbreed. It can be a confound in genetic studies that assume a homogenous population if the ancestral genomes are present in different proportions in cases and controls., 12

Allele An allele is one of the different possible forms at a locus. , 5

Alternative hypothesis In statistical hypothesis testing, the alternative hypothesis is the opposite of the null. It is assumed false for the purposes of the test. , 42

Attribute In an instance, an attribute is a measured or computed variable. An example of a computed variable would be the variables representing pairs of SNPs that are used by Random Forest., 10

Background rate The background rate of disease with respect to a single disease cause is the probability of disease in a population without taking the cause into effect. It covers disease caused by environmental effects or genetic causes that were not measured., 38

Bonferroni correction A correction to significance tests to account for the testing of multiple hypotheses. The idea is that 1 in 20 tests at a significance level of $\alpha = 0.05$ leads to a rejection of H_0 provided that H_0 is true all 20 times. To correct for the $n\alpha$ tests that will differ significantly from H_0 if n tests are performed, the Bonferroni correction tests at a significance level of α/n, a much more stringent test for large n. , 66

Bootstrap sampling Bootstrapping is a means of estimating an unknown population parameter by examining an approximate distribution gotten by repeatedly sampling with replacement from observed data in order to simulate new observations. In order to work, the population must be independent and identically distributed., 17

Chi-square ($\chi^2_{1d.f.}$) The $\chi^2_{1d.f.}$ distribution is equivalent to the square of the standard normal distribution. With one degree of freedom, the mean is 1 and variance is 2. While the chi-square is defined for higher degrees of freedom, we only use 1 d.f. in this thesis., 8

Cross-validation (CV) A technique for validating machine learning algorithms. Data is subdivided into subsets called folds. The learning algorithm is trained on all but one of the folds and the remaining fold is classified using the resulting learner. This is repeated once per fold so that every element is out-of-bag in one learner. In a single iteration, the folds that are used to build the model are called the training set while the fold that is classified is called the test set., 22

Deoxyriboncleic acid (DNA) A molecule that contains the instructions for proper function and development of organisms. DNA exists as a pair of macromolecules that are united by hydrogen bonds and wound into a double helix., 3

Dominant association The detection of an association between one or more copies of an allele and phenotype. This corresponds to a dominant model, in which one or two copies of an allele convey the same risk while no copy conveys a different risk probability., 4

Epistasis Epistasis is defined in this document as a non-additive relationship between a set of SNPs and a phenotype. Equivalently, this entails a deviation

from independence with respect to phenotype. Originally, epistasis was defined to require that one SNP or gene suppressed the effect of another, but it is important to note that epistasis is equivalent to interaction in this thesis., 43

Genome-wide association study (GWAS) A study of genetic variation across an entire genome with the intention of discovering genetic locations that are associated with phenotypes., 2

Genotype The genetic composition of an organism or individual. Could refer to the composition across the entire genome or at specific bases depending on the context., 7

Hyperplane A hyperplane is a function of the form $f(\mathbf{x}) = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n = 0$., 32

Instance A data element. In our data, an instance is an individual., 10

Linearly separable Two classes of data in $\mathbf{R}^n$ are linearly separable if one can draw a hyperplane that perfectly divides the classes., 33

Linkage disequilibrium A statistical association between two loci that are in close enough physical proximity that the probability of recombination between them is less than 0.50. , 4

Minor allele frequency (MAF) For a population in which two alleles are present at a given locus, the minor allele is the less common variant and the MAF is the frequency of occurence of the less common variant. MAFs range between 0 and 0.50., 39

Null hypothesis (H_0) A hypothesis that could be shown false by a test on observable data. Generally, H_0 is formed and the probability that it is true is calculated given an observation (see p-value.) , 5

Out-of-bag An instance is out-of-bag in a CV test if it is in the test rather than training set., 22

P-value In statistical hypothesis testing, a p-value p denotes the probability of observing a statistic that is as extreme or more extreme that the observed statistic given that the null hypothesis is true. For example, if one is testing the null hypothesis that a coin is fair, one might calculate the probability of observing 18 of one side in 20 flips of the coin. A p-value would give the probability of observing $18, 19$, or 20 of the same side in 20 flips of the coin given that the coin is fair., 27

Penetrance The penetrance of a genoype in regard to a phenotype is the proportion of a population with a given genotype that displays the phenotype., 38

Polymorphism In genetics, polymorphism is the existance of more than one genetic form at the same locus at a rate that can not be accounted for by recurrent mutation (i.e. mutation in the current generation alone.) , 5

Positive predictive value (PPV) The probability that the null hypothesis is false given that it was rejected., 7

Power The probability of rejecting a null hypothesis given that it is false., 7

Radial kernel In an SVM, a radial kernel measures distance from the origin using the function $r(x_1, x_2) = exp(\gamma \times \text{norm}(x_1 - x_2)^2)$., 33

Recessive association The detection of an association between two copies of an allele and phenotype (but not between one or two copies). This corresponds to a recessive model, in which two copies of an allele convey the same risk while no copy or one copy conveys a different risk probability., 4

Recursive feature elimination (RFE) A feature detection technique where the set of possible features is refined by removing a single feature from the set at each step in the algorithm. Features that are selected for removal are least relevant for classification. RFE avoids the tradeoff between greedy approaches or heuristics in creating a data set by adding features. [8, 37], 34

Ribonucleic acid (RNA) A nucleic acid that contains nucleotides and is transcribed from DNA. RNA does not typically form hydrogen bonds with itself, but instead exists as a single strand., 3

Single nucleotide polymorphism A SNP is a single location in the genome where the allele present differs among members of the same species., 4

Sporadic rate Many diseases have multiple causes, and a GWAS might not detect all causes. For instance, environmental factors and unmeasured genetic effects may play a role. We account for the fact that a measured locus does not determine all cases of a disease by including a sporadic rate in our genetic models. The sporadic rate applies to all individuals., 39

Statistical association Association between a set of variables and a response variable is defined as occuring when there is dependence between any variable and the response variable. Association between x_1, x_2 and y exists if β_1 or β_2 is non-zero in the equation $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2$., 43

Statistical interaction An additive relationship between x_1, x_2 and y is captured

Bibliography

[1] Alan Agresti, *Categorical data analysis*, Wiley, 1990.

[2] Leo Breiman, *Bagging predictors*, Machine Learning, 1996, pp. 123–140.

[3] Leo Breiman, *Random forests*, Machine Learning **45** (2001), 5–32.

[4] Leo Breiman and A. Cutler, *Random forests. version 4.0.*, Dec 2009, `http://www.stat.berkeley.edu/users/breiman/RandomForests/`.

[5] Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen, *Classification and regression trees*, Chapman and Hall, CRC, 1984.

[6] Alexandre Bureau, Josée Dupuis, Kathleen Falls, Kathryn L. Lunetta, Brooke Hayward, Tim P. Keith, and Paul Van Eerdewegh, *Identifying snps predictive of phenotype using random forests*, Genetic Epidemiology **28** (2005), 171–182.

[7] CC Chang and CJ Lin, *Libsvm: A library for support vector machines*, Dec 2009, `http://www.csie.ntu.edu.tw/cjlin/libsvm`.

[8] Shyh-Huei Chen, Jielin Sun, Latchezar Dimitrov, Aubrey R. Turner, Tamara S. Adams, Deborah A. Meyers, Bao-Li Chang, S. Lilly Zheng, Henrik Grönberg, Jianfeng Xu, and Fang-Chi Hsu, *A support vector machine approach for detecting gene-gene interaction*, Genetic Epidemiology **32** (2008), 152–167.

[9] N. Christianini and M.W. Hahn, *Introduction to computational genetics: A case studies approach*, Cambridge UP, 2007.

[10] N. Christianini and J. Shawe-Taylor, *Support vector machines and other kernel-based learning methods*, Cambridge UP, 2000.

[11] Heather J. Cordell, *Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans*, Human Molecular Genetics **11** (2002), 2463–2468.

[12] H.J. Cordell, *Detecting gene-gene interactions that underlie human diseases*, Nature Genetics **10** (2009), 392–404.

[13] Yoav Freund and Llew Mason, *The alternating decision tree learning algorithm*, In Machine Learning: Proceedings of the Sixteenth International Conference, Morgan Kaufmann, 1999, pp. 124–133.

[14] Phillip Good, *Permutation tests*, Springer, 1994.

[15] Casey S. Greene, Daniel S. Himmelstein, Heather H Nelson, Karl T. Kelsey, Scott M Williams, Angeline S. Andrew, Margaret R. Karagas, and Jason H. Moore, *Enabling personal genomics with an explicit test of epistasis*, Pac Symp Biocomp (2010), 327–36.

[16] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten, *The weka data mining software: An update*, SIGKDD Explorations **11** (2009).

[17] JB Harley, ME arcon Riquelme, LA Criswell, CO Jacob, RP Kimberly, KL Moser, BP Tsao, TJ Vyse, CD Langefeld, J Divers, W Wang, MC Marion, and A Williams, *Genome-wide association scan in women with systemic lupus erythematosus identifies susceptibility variants in itgam, pxk, kiaa1542 and other loci.*, Nat Genet. **40** (2008 Feb), 204–210.

[18] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning*, Springer, 2001.

[19] David W. Hosmer and Stanley Lemeshow, *Applied logistic regression*, Wiley-Interscience, 2000.

[20] C.D. Langefeld and T.E. Fingerlin, *Association methods in human genetics*, Topics in Biostatistics.

[21] K.Y. Liu, J. Lin, X. Zhou, and S.T. Wong, *Boosting alternating decision trees modeling of disease trait information*, BMC Genet. **6(Suppl. 1)** (2005), S132–S138.

[22] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, *Introduction to information retrieval*, ch. 8, Cambridge UP, 2008.

[23] David J. Miller, Yanxin Zhang, Guoqiang Yu, Yongmei Liu, Li Chen, Carl D. Langefeld, David Herrington, and Yue Wang, *An algorithm for learning maximum entropy probability models of disease risk that efficiently searches and sparingly encodes multilocus genomic interactions*, Bioinformatics (2009), btp435.

[24] Jason H. Moore, Joshua C. Gilbert, Chia-Ti Tsai, Fu-Tien Chiang, Todd Holden, Nate Barney, and Bill C. White, *A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility*, Journal of Theoretical Biology **241** (2006), 252–261.

[25] Jason H. Moore and Scott M. Williams, *New strategies for identifying gene-gene interactions in hypertension*, Annals of Medicine **34** (2002), 88–95.

[26] Jason H. Moore and Scott M. Williams, *Traversing the conceptual divide between biological and statistical epistatis: system biology and a more modern synthesis*, BioEssays **27** (2005), 637–646.

[27] M.R. Nelson, S.L.R. Kardia, R.E. Ferrel, and C.F. Sing, *A combinatorial partitioning method to identify multilocus genotypic partitions that predict quantitative trait variation*, Genome Res. **11** (2001), 458–470.

[28] Douglas K. Owens and Harold C. Sox, *Biomedical decision making: Probabilistic clinical reasoning*, Biomedical Informatics.

[29] Bernhard Pfahringer, Geoffrey Holmes, and Richard Kirkby, *Optimizing the induction of alternating decision trees*, Proceedings of the Fifth Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining (PAKDD2001), 2001, pp. 477–487.

[30] Marylyn D. Ritchie, Lance W. Hahn, and Jason H. Moore, *Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity*, Genetic Epidemiology **24** (2003), 150–157.

[31] Marylyn D. Ritchie, Lance W. Hahn, Nady Roodi, L. Renee Baily, William D. Dupont, Fritz F. Parl, and Jason H. Moore, *Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer*, Am. J. Hum. Genet. **69** (2001), 138–147.

[32] Matt L. Stiegert, Joshua D. Grab, Richard T. Guy, and Carl D. Langefeld., *Snpgwa version 4.2*, http://www.phs.wfubmc.edu/public/bios/gene/downloads.cfm, 2003–2010.

[33] Carolin Strobl, Anne-Laure Boulesteix, Achim Zeileis, and Torsten Hothorn, *Bias in random forest variable importance measures: Illustrations, sources, and a solution*, BMC Bioinformatics **8** (2007).

[34] T. A. Thornton-Wells, J. H. Moore, and J. L. Haines, *Genetics, statistics and human disease: analytical retooling for complexity.*, Trends in genetics : TIG **20** (2004), no. 12, 640–647.

[35] John Wilder Tukey, *Exploratory data analysis*, Addison-Wesley, 1977.

[36] Digna R. Velez, Bill C. White, Alison A. Motsinger, William S. Bush, Marylyn D. Ritchie, Scott M. Williams, and Jason H. Moore, *A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction*, Knowledge Information Systems **31** (2004), 306–315.

[37] Ian H. Witten and Eibe Frank, *Data mining: Practical machine learning tools and techniques*, Morgan Kaufmann, 2005.

[38] X. Wu, V. Kumar J. Ross Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinback, David J. Hand, and Dan Steinberg, *Top 10 algorithms in data mining*, Knowledge Information Systems **14** (2008), 1–37.

Vita

Richard T. Guy

BORN:	November 3, 1984, Milwaukee, WI
UNDERGRADUATE STUDY:	Appalachian State University Boone, North Carolin B.A. *Summa Cum Laude*, Mathematics, Philosophy and Religion, May 2007
GRADUATE STUDY:	Wake Forest University Winston-Salem, North Carolina M.A., Mathematics, May 2009 M.S., Computer Science, May 2010
	University of Toronto Toronto, Ontario, Canada Ph.D., Computer Science, starting in August, 2010

Papers

[12] P.S. Ramos, J.T. Ziegler, M.E. Comeau, A.H. Williams, R.T. Guy, C.J. Lessard, L.A. Criswell, P.M. Gaffney, J.A. Kelly, K.M. Kaufman, R. Zidovetzki, C.O. Jacob, R.P. Kimberly, B.P. Tsao, T.J. Vyse, J.B. Harley, M.E. Alarcón-Riquelme, C.D. Langefeld, and K.L. Moser, "Genetic Analyses of Interferon Pathway-Related Genes Reveals Multiple New Loci Associated with Systemic Lupus Erythematosus (SLE)" In final revision; submitting to The American Journal of Human Genetics.

[11] R.T. Guy, W. Wang, M. Marion, P. Ramos, K. Wilson, T. Howard, C.D. Langefeld "SNPDOC: A Web Based SNP Selection Utility," in preparation.

[10] A. Williams, R.T. Guy, C. Langefeld, T. Fingerlin, "Multiple Imputation Strategies for Haplotype Association Analyses in General Pedigrees" in preparation.

[9] Q. Zhang, R.T. Guy and R.J. Plemmons, "Matrix Structures and Parallel Algorithms for Image Superresolution Reconstruction," accepted by the SIAM Journal of Matrix Analysis and Applications.

[8] K. S. Berenhaut, R. T. Guy and C. L. Barrett, "Globally asymptotic behavior for minimum difference equations," accepted by the International Journal of Difference Equations.

[7] R. T. Guy, M. Misiurewicz, "Euler Approximations Can Destroy Unbounded Solutions." Accepted for PODE09 proceedings, special issue of Fasciculi Mathematici.

[6] M. Mirotznik, S. Mathews, R. Plemmons, P. Pauca, T. Torgersen, R. Barnard, R.T. Guy, B. Gray, Q. Zhang, J. van der Gracht, C. Peterson, M. Bodnar, S. Prasad, "A Practical Enhanced-Resolution Integrated Optical-Digital Imaging Camera (PERIODIC)" in Proc. SPIE Conference on Defense Security and Sensing, April, 2009.

[5] K. S. Berenhaut, R. T. Guy, "Symmetric Functions and Difference Equations with Asymptotically Period-Two Solutions," in press. (International Journal of Difference Equations.)

[4] K. S. Berenhaut, R. T. Guy, "Periodicity and boundedness for the integer solutions to a minimum-delay difference equation," in press. (Journal of Difference Equations and Applications)

[3] K. S. Berenhaut, R. T. Guy, and N. G. Vish, "An Optimal Bound for Inverses of Triangular Matrices with Monotone Entries," accepted by the Journal of Linear and Multi-linear algebra.

[2] K. S. Berenhaut, R. T. Guy, and N. G. Vish, "A 1-norm bound for inverses of triangular matrices with monotone entries," Banach J. Math. Anal. 2 (2008), no. 1, 112{121}.

[1] J. Glenn, K. Creehan, E. Van Aken, R.T. Guy, "Dynamic Value Stream Mapping Software," Patent Application Number: 60/822,558, Filed by Dorrity and Man-

ning, August 2007.

Presentations

[10] "Applications of number theory to asymptotic behavior of solutions of difference equations." Invited Talk at Joint Mathematical Meetings in San Francisco, January, 2010.

[9] "Some recent results on minimum-delay difference equations" Invited Talk at AMS Regional Meeting at The Pennsylvania State University on October 24, 2009.

[8] "Periodicities in Composition-Delay Difference Equations," Progress on Difference Equations 2009, Bedlewo, Poland, May, 2009.

[7] "Challenges and Opportunities in SNP Data Analysis," Math Club lecture, Wake Forest University, April, 2009.

[6] "Some new results on Composition-Delay Equations with Asymptotically Periodic Solutions" poster presented at Graduate Research Poster Session, Wake Forest University, April 2009.

[5] "Symmetric Functions and Difference Equations with Asymptotically Period-Two Solutions" presented January, 2009, Joint Mathematics Meeting (JMM), Washington D.C.

[4] "The periodic character of some rational difference equations of arbitrary order with truncation (Preliminary report)" presented January, 2008, JMM, San Diego, CA

[3] "Diagonalization and the Liar" presented April of 2007 at National Conference on Undergraduate Research, Dominican University, San Raphael, CA

[2] "A Revaluation of Marriage: and the Two Shall Remain Two" presented on Oct 14, 2006 at the Southeastern Undergraduate Philosophy Conference at UNCA, Asheville, NC

[1] "The Humanist Lemma: A Means of Discourse" presented February of 2006 at the Annual Meeting of the North Carolina Philosophical Society, Columbia, SC

Service

Chair of session "Contributed Papers in Difference and Functional Equations" at JMM 2009, Washington D.C.

Graduate President, Pi Mu Epsilon, Wake Forest University, 2008-2009

Senator, Student Government Association, Appalachian State University, 2005-2006

Honors and Awards

Gordon A. Melson Outstanding Graduate Student (2009) Graduate Research Day Poster Competition winner (2009) and runner up (2010), Wake Forest University

Upsilon Pi Epsilon inductee.

Pi Mu Epsilon inductee.

Senior Class Representative and podium speaker for May Commencement, 2007, Appalachian State University

Chancellor's List (GPA > 3.85) every semester at Appalachian State University

Academic Excellence Award, 2006, 2007, Appalachian State University

Top Senior Award, Department of Philosophy and Religion, Appalachian State University, 2006, 2007

Academic Organizations

Association for Computing Machinery

International Society for Difference Equations (Sponsoring member)

American Mathematical Society