IN SILICO EVALUATION OF DNA-POOLED ALLELOTYPING VERSUS INDIVIDUAL GENOTYPING FOR GENOME-WIDE ASSOCIATION STUDIES OF COMPLEX DISEASE

By

Siddharth Pratap

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

Biomedical Informatics

August, 2007

Nashville, Tennessee

Approved:

Professor Shawn Levy

Professor Dan Masys

Professor Jay Snoddy

Professor Scott Williams

DEDICATION

Dedicated to my wife Sanju and my sons Vishnu and Arjun.
You are the reasons why I smile, thank you for everything.

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Shawn Levy as well as my advisory committee, Dan Masys, Jay Snoddy, and Scott Williams for their wonderful guidance in this project as well as their very helpful advice both professional and beyond.

I would like to thank the National Library of Medicine and the National Institutes of Health, whose generous support made this research possible (Training Grant T15 007450-03).

I would like to thank the Department of Biomedical Informatics for giving me this opportunity to continue as well as expand my development as a scientist.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

With the recent advances in the human genome project and with the International HapMap project adding information on millions of SNPs to the knowledge base, the feasibility of applying large-scale and even genome-wide association studies for complex disease and quantitative traits is fast approaching. Recent advances in SNP genotyping techniques now support genome-wide profiling of hundreds of thousands, even millions, of SNPs in parallel with proven accuracy. These technologies allow for a hypothesis generating approach to identifying disease associated alleles without *a priori* knowledge or candidate genes or regions. One obstacle in the progression towards whole-genome association studies for complex diseases and quantitative traits is the high cost of projects that are appropriately statistically powered for association analysis. For example, if disease associated alleles have a **minor allele frequency (MAF)** of less than 0.1 and an effect size less than an odds ratio of 1.3, then the sample size needed for an association study with a statistical power of >80% at a significance level of p-value < $1e^{-6}$ would be more than 10,000 cases and an equal number of controls [1]. While the natures of MAF and odds ratios are not comprehensively characterized for complex diseases and quantitative traits, the sample size requirements in the above example are not extreme. In a 2007 review by Couzin and Kaiser, the results of several **genome-wide association (GWA)** studies are listed [2]. Table 1 summarizes these results with respect to sample size, number of disease associated variants, and the increased disease risk attributed to the variants.

Table 1: Review of Genome-Wide Association study results:

Selected Genome-Wide Scan Results

DISEASE	PUBLICATION DATE	SAMPLE SIZE*	GENES OR VARIANTS FOUND	APPROXIMATE INCREASED RISK FOR HOMOZYGOTES†
Macular degeneration	2005	1700	1 new gene	400% to 600%
Inflammatory bowel disease	2006	4500	1 new gene	120%
Prostate cancer	2007	17,500	2 variants in same region (1 new)	123%
Obesity	2007	38,700	1 new gene	67%
Type 2 diabetes	2007	32,500	9 variants (3 new)	80%
Heart disease	2007	41,600	1 new variant	25% to 40%

* Cases and controls including replicates. † For highest risk variant.

Couzin and Kaiser [12]

In another review of currently known quantitative trait loci (QTL), approximately half of the candidate causal variants had MAFs of less than 0.05 [3], and the odds ratios and relative risks of known disease associated variants often occur in the range between 1.1 and 1.5 [4, 5]. Therefore, individually genotyping the thousands of individuals necessary to achieve proper statistical results could rapidly become a multi-million dollar undertaking, prohibitive for all but the most highly funded labs.

A potential solution to the prohibitive cost of individual genotyping is to combine genomic DNA from case individuals and an equal number of control individuals and to genotype the pooled DNA. Pooling designs for association studies of complex disease such as Alzheimer's [6], sudden infant death with dysgenesis of the testes [7], and mild mental impairment [8] have had promising results. However, using pooling as a general approach to identify complex disease associated genes remains controversial.

While pooling has obvious advantages in that fewer genotyping assays result in substantial cost and time savings, questions arise as to the ability to detect the causal or associated alleles with small effect size and/or low frequencies in the pooled samples versus the individually genotyped samples. Further, the characteristics of information obtained relative to individual genotyping need to be explored in order to determine the overall validity of DNA pooling for genome-wide association analysis. In this study, we propose to examine the feasibility of a pooling approach for whole-genome association studies of complex disease. Using a simulated genome dataset, we will compare and evaluate the advantages and disadvantages of DNA pooling versus individual genotyping.

The genetic architectures of common disease are the results of complex interactions from multiple alleles, as well as gene-environment and gene-gene interactions. The allelic spectrum of complex disease can be modeled as interplay between the number of disease variants, the risks that these variants confer, and the frequencies with which they occur in the population. Taking these factors into account, the statistics of association studies and the power which they can bring to that study will emerge.

Two models of the characteristic patterns of common or complex diseases have been proposed. First, the common disease/common variant hypothesis suggests that a disease results from the ***multiplicative*** action of several common variants. Unrelated affected individuals have a significant proportion of disease alleles in common, hence the term "common variant" [3, 9]. In the case of this multiplicative model, the combined effect of multiple disease associated alleles contributes exponentially as opposed to linearly. In contrast, the classical disease heterogeneity hypothesis (or multiple rare-variant hypothesis) suggests that disease susceptibility is due to rare and distinct variants in different individuals which contribute additively [10]. The ***additive model*** of complex disease states that each disease associated allele contributes equally to the overall disease risk. Studies have shown evidence for an additive characteristic spectrum in some cancers [11] and type 2 diabetes [12-14] and a multiplicative effect in type 1 diabetes [15].

There are two main methodologies for identifying complex disease genes. The first are candidate gene studies using either association or resequencing protocols. Association candidate gene studies have the advantage of being relatively inexpensive

and are able to detect alleles with modest effect size, given that these alleles are common in the population (MAF>0.05). However, both association and resequencing candidate gene studies require prior information about gene function. The second approach for identifying common disease variants is to use genome-wide studies: either linkage mapping or genome-wide association. Although genome-wide linkage analysis has been successful, its greatest power lies in identifying Mendelian diseases which are characteristically monegenic and highly penetrant [16]. In contrast, complex diseases are typically the result of multiple causal loci each with low penetrance. Thus, linkage studies for complex disease have had only limited success and limited reproducibility. For example, disease variants have been found in inflammatory bowel disease, which account for a two-fold increase in risk in siblings [17-20]. Yet it has been shown that a greater than a thirty-fold risk increase exists overall for this disease, pointing to the existence of other genes not resolved by the linkage study design [21]. The power of linkage analysis decreases sharply as effect of complex disease genes becomes less penetrant [22-25]. Linkage analysis is more powerful than association analysis for identifying rare, highly penetrant alleles, but association analysis is expected to be more powerful for identifying complex disease alleles with modest disease risks [22]. Because linkage analysis is done in families, the resulting regions of linkage correlating with disease are relatively large, often on the order of 10cM or more (~10 million base pairs). Therefore, extensive candidate gene analysis (either by resequencing or association studies) must follow in order to find the causal genes within the linked region [16].

Genome-wide association studies offer the advantage of not needing prior information about linkage or of candidate genes in order to attain associated alleles in

complex diseases. In contrast to candidate gene based studies, genome-wide association will be unbiased and, if designed properly, also fairly comprehensive in terms of extracting a significant portion of the genomic variation. A crucial advance made possible through the HapMap project is the characterization of **linkage disequilibrium (LD)** patterns on a genome-wide scale [26, 27]. This is important for indirect association methods that use markers selected on the basis of LD. To be useful, markers tested for association must either be the causal allele (direct association) or in LD with the causal allele (indirect association) [28, 29]. Roughly 70-80% of the genome falls into segments of strong LD and variants with high LD are strongly correlated with each other [30]. Most of the ~11 million common SNPs (MAF>0.01) in the genome have groups of neighbors that are all nearly perfectly correlated with each other [28]. One SNP can thereby serve as a proxy for many others in an association screen. Overall, this suggests that if patterns of LD are known for a given region, a few tagSNPs can be chosen which, either individually or in multimarker combinations (haplotypes), capture most of the common variation within the region [31, 32]. Thus, it has been proposed that much of the common variation in the genome can be found by genotyping a few hundred thousand "well-chosen" SNPs [33, 34]. For the remaining fraction of the genome that is not in LD (20-30%), higher densities of variants must be typed. Such a scan is estimated to require two hundred thousand to one million markers to achieve a reasonable likelihood that any common SNP in the genome is usefully associated with at least one tagSNP [35]. Current genotyping technologies make use of a "tagSNP" centered approach such as Illumina's "HumanHap" series of whole-genome beadchips.

Advantages and disadvantages of pooled genotyping for genome-wide association analysis:

"In their simplest form, association studies compare the frequency of alleles or genotypes of a particular variant between disease cases and controls." (Wang 2005 [1])

The pooling approach lends itself well to association studies as the allele frequencies themselves are the results and outputs of pooled genotyping. The major benefit of DNA pooling is that it reduces the amount of genotyping that is required to estimate allele frequencies, as fewer SNP chips are utilized. Additionally, as a direct result of less genotyping, the time of analysis is also reduced. Thus, the efficiency of pooling is directly correlated with the number of samples pooled. These pools could be constituted from cases and controls for a disease trait, or from individuals with values at the two extremes of a quantitative trait. Studies have been done where the optimal tails of quantitative trait distributions have been derived [36]. In a symmetrical design, taking the top and bottom 27% of trait distributions and pooling them extracts 80% of the total information compared to individual genotyping for common alleles with an additive effect [36]. These results do not generalize to recessive or rare alleles however, in this case an asymmetrical design is needed [37].

The degree of power conferred by pooling is highly influenced by the effect size of the causal variants. Alleles which have large relative risks will directly lead to large differences in allele frequency estimates between cases and controls. However, there will likely be small differences between case and control allele frequencies in complex disease because of the small increase in relative risks resulting from modest risk alleles.

The ability to detect the real allele frequency difference is dependent on the differences in allele frequencies between cases and controls, and the errors and variances in estimating them. This error in allele frequency estimates for pooling comes from two main sources: sampling error and measurement error [38]. Measurement error results from "poor quality DNA" which has degraded, or from platform genotyping error. Measurement error is present in both individual and pooled genotyping. Sampling error results from uneven DNA contributions to the pool from individuals and platform errors in allele frequency estimation. Proper DNA mixing can be controlled by careful measurement and mixing of equal amounts of DNA into the pool with the use of robotics and highly sensitive DNA quantitation protocols such as Picogreen. Increasing sample size reduces sampling error, but not measurement error. Estimated allele frequencies from pools have consistently been found to have standard deviations between 2-4% due to errors resulting from the genotyping platforms [38]. In the case of complex disease where modest-risk alleles with small effect size are likely, even small allele frequency estimate errors can dramatically reduce the power of a pooled methodology. In a simulation study, it was shown that the variance resulting from sampling error was potentially more significant than the variance resulting from measurement error [39]. Thus, pooling designs have the disadvantage of having an additional source of error, namely sampling error, which is not present in individual genotyping.

One major disadvantage of pooling is that the haplotype information is not resolvable as only allele frequencies are determined by pooled genotyping. This actually makes the term "pooled genotyping" a bit misleading. Pooled "allelotyping" has been suggested as the allele frequencies are the results from a pooled experiment. There are

special circumstances where haplotype information can be obtained from a pooled sample. For pool sizes of less than 10 individuals per pool, it is possible to determine haplotype frequency estimates using the expectation-maximization (EM) algorithm [40]. However, these are not true haplotypes in terms of the genotype information and their chromosomal location. Rather, the frequency estimates of the haplotypes in the sense of the population.

A potentially useful outcome of the loss of haplotype information by pooling may be realized. This inability to identify individual haplotypes from a pooled DNA sample effectively de-indentifies the individuals in the pooled groups.

The high number of genotype tests involved in a whole-genome association study will certainly lead to many false positives. If a liberal significance threshold of (p-value <0.05) is used, approximately 5,000 false positives will result from a 100,000 SNP gene chip. Applying a more stringent p-value cutoff such as $5x10^{-7}$, (P-value = 0.05) which has been Bonferroni corrected for 100,000 independent tests, will lower the number of false positives but also risks making the testing too stringent and may lead to missed associations. The ultimate choice of stringency cutoff is best left to the circumstances of the experiment. This is an area of active research. Three likely sources of false positives are statistical fluctuations from random noise, underlying systematic biases due to study design, and technical artifacts.

Conclusions of introduction:

This study is firstly an exploration to address the feasibility of DNA pooling for whole-genome association analysis of complex disease. While previous research has

shown encouraging gains in efficiency by pooling for several diseases (including Alzheimers') [5,6], this may be specific to the nature of disease in question and not generalizable. There have also been well structured designs to explore the impact of genotyping and experimental errors in pooling designs in terms of the change in sample size needed to achieve valid statistical results [39]. The effect of genotyping error on the haplotype frequency estimates has also been explored [40]. To our knowledge, a comprehensive analysis of the tradeoffs involved in a pooling design versus individual genotyping has not been systematically addressed. Additionally, there have been limitations at generating complex disease models which are good approximates of the diseases. Put simply, the previous generations of genotype simulation software does not allow for models with enough complexity to sufficiently represent a complex disease. This study aims to create and evaluate a more complex model of common disease. The International Haplotype Mapping project (HAPMAP) has defined the genome with a much high degree of resolution than ever before [26, 27, 35, 41]. Commercially available genotyping platform technologies allow genotype scanning at progressively higher densities. The convergence of technology and knowledge presents an opportunity to begin true genome-wide association analysis.

CHAPTER II

COMPLEX DISEASE MODEL MATHEMATICS

Pooled genotyping association testing statistic:

The appropriate test for this two-pool design would be to consider the magnitude of the difference between the allele frequency estimates of the two pools in relation to its variance. The standard Pearson's chi-squared test assumes that the variance of the difference in allele frequency estimates is determined entirely by sampling variation alone. This does not apply to pooling because the variance will be inflated by sampling and allele frequency measurement errors that are specific to DNA-pooling studies [36, 37, 42-44]. These additional errors can potentially increase the number of false-positive association findings.

Equation 1 shows the appropriate test statistic for allelic association for two independent pools.

Equation 1: *Pooled genotyping association testing statistic*

$$z^2 = \frac{(p_1 - p_2)^2}{V_1 + V_2}$$

Equation 1 states that pooled association statistic testing is calculated by taking the difference in allele frequency measurements on the two pools (p_1 and p_2), squaring that difference, and dividing by the sum of two variances ($V_1 + V_2$). V_1 is the variance in allele frequency estimates due to platform measurement error. V_2 is variance in allele

frequency estimates due to sampling error. Measurement error variance (V_1) of the allele frequencies results mainly from the genotyping platform itself [38] or by degraded "bad quality" DNA resulting in genotyping errors. Sampling error variation (V_2) is attributed mainly to unequal amounts of DNA used to construct the pools by imprecise quantification and inaccurate mixing, and by signal measurement error by the genotyping platform. Both of these sources of sampling error skew the allele frequency estimates. Under the null hypothesis and with a two pooled design, Z^2 is approximately distributed as a chi square distribution with one degree of freedom [37], given a sample size of 100 or more [45].

Due to the nature of complex disease alleles having a modest effect size, it has been shown that taking the mean association test statistic for consecutive markers can be more informative than single point association. Using a moving windows approach where consecutive marker groups of varying window size are tested takes advantage of the fact that markers in LD will be reciprocally predictive of each other. If markers are not in LD, permutation testing can be used in order to ascertain the significance of a window group [7].

Complex disease model characteristics:

There are two thoughts as to how the allelic profile of complex disease can be modeled. These two models are not so much competing views as they are complementary possibilities of how complex disease might look at the genetic level. Firstly, the **additive model** hypothesizes that each disease associated allele contributes roughly equally in terms of increased disease risk, and that numerous instances of these

disease associated alleles have an additive effect [1]. This additive model is also termed the disease heterogeneity model as the disease risk increases linearly with the number of risk alleles present.

The contrasting model proposed for complex disease is the **multiplicative model**. It states that disease risk increases exponentially with increasing numbers of disease associated variants present. Evidence for both additive and multiplicative types of complex disease allelic spectra exists in the field. For example, type 2 diabetes has been shown to display an additive characteristic of disease contribution by several alleles of modest effect [14, 46, 47] while type 1 diabetes may be multiplicative[12]. In order to build these two models into our genotype simulation package (genomeSIM), penetrance tables were constructed in order to represent the additive and multiplicative models. The disease risk calculations for both the additive and multiplicative models have been set to include a phenocopy rate of 10%. This is the disease risk due to environmental factors, as well as other non-genetic factors. A graph of the two types of complex disease models taken from a review by Wang et. al. is shown in Figure 1. It relates the probability of disease between the two model types versus genetic relatedness.

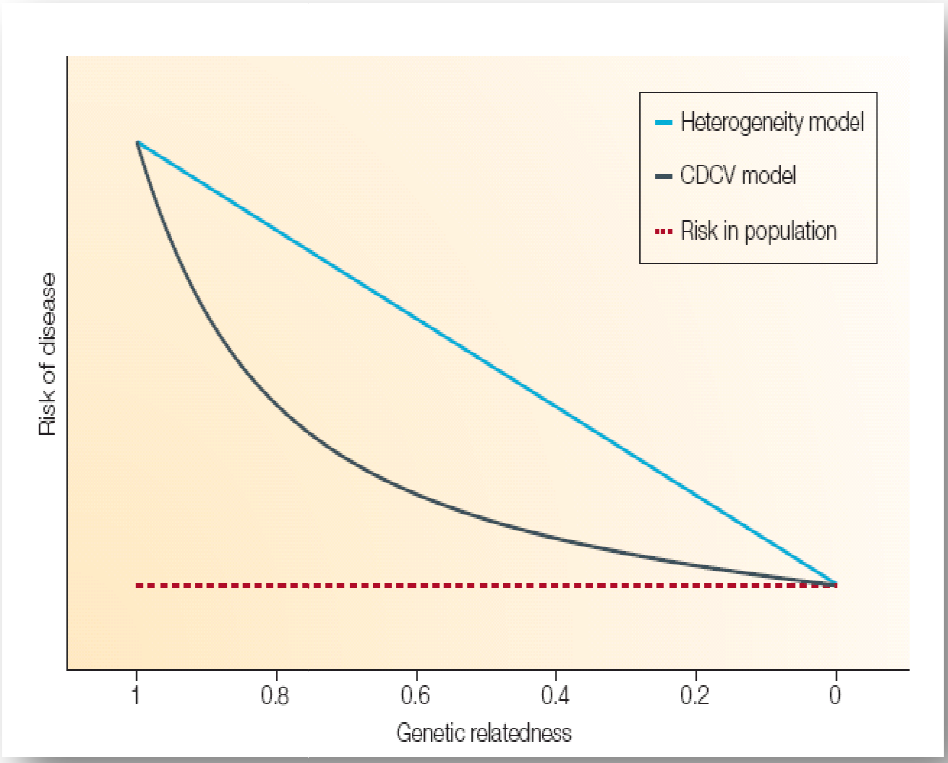


Figure 1: Schematic representation of allelic spectrum of Heterogeneity (Additive) and Common Disease Common Variant (Multiplicative) complex disease models

Wang et. al. [1]

Complex disease models for construction of simulated genotypes:

The method of calculating the disease probability values in the penetrance tables was derived using a generalized relative risk model for complex disease proposed by Risch and Teng [48]. This genetic model serves as a general framework for assigning a disease penetrance to a corresponding genotype. If a disease locus contains alleles "D" for wild-type and "d" for a disease associated variant, then the penetrance associated with the disease genotypes can be represented as DD = *f0*, Dd = *f1*, and dd = *f2*. Given this representation, disease models can then be expressed in terms of relative penetrance or genotypic risk ratios [22]. It follows that if the risk associated with the wild-type genotype "DD" is fixed at *f0*=1, then the risk associated with a single diseased allele heterozygote, ("Dd" = *f1*), can be expressed relative to the *f0*. Further, the risk associated with two disease alleles, ("dd" = *f2*), can also be expressed relative to the *f1* and the *f0*. The additive and multiplicative complex disease models can then be defined in terms of the *f0*, *f1*, and *f2* penetrance values. For the additive model the formula [$f2 = 2(f1) - f0$] describes the relationship. A multiplicative model can be represented by [$f2 = f1^2$][48]. These models are for a single locus and allow for representation of a three state model, namely [DD = *f0*; Dd = *f1*; and dd = *f2*]. So an additive model with a relative risk of 5 might be defined by the following parameters: DD (*f0* = 0.1) corresponding to wild type genotype with a 10% probability of having the disease, Dd (*f1* = 0.5) corresponding to a 5 fold increase in disease probability given the presence of one disease associated allele, and if both alleles are disease variants, dd (*f2* = 0.9) calculated by application of the additive complex disease model formula [$f2 = 2(f1)-f0$] -> [$f2 = 2(0.5)-0.1$] -> $f2 = 0.9$.

Our evaluation applied the Risch and Teng complex disease formulas to a three locus model involved a stepwise expansion of the penetrance function relationships to the genotypes and consequent expansion of the model formulas themselves. A one locus model will have 3 possible states corresponding to [DD = *f0*, Dd = *f1*, dd=*f2*]. Expanding this to a two locus biallelic model (genes "A" and "B") will give 9 possible genotypic states [AABB, AABb, AAbb, AaBB, AaBb, Aabb, aaBB, aaBb, aabb]. These 9 states are the result of the 3 possible genotype states for each of the 2 loci. The combinatorics result from the number of possible states for each bi-allelic locus (3) raised to the number of loci (2), [$3^2 = 9$]. Accordingly, a 3 locus model will have $3^3 = 27$ possible genotype states. For this study, each of the model states is represented analogous to the Risch and Teng model as a genotype and corresponding penetrance probability. Penetrance Table 1 shows a penetrance function in which the additive and multiplicative disease probabilities were calculated for the 27 genotypes of each model.

Table 2: Penetrance Table 1 for Relative Risk Range 1.08 to 1.50

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				**Disease Risk (Probability)**	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.1083	0.1070
AA	Bb	CC	AABbCC	0.1083	0.1070
AA	BB	Cc	AABBCc	0.1083	0.1070
aa	BB	CC	aaBBCC	0.1167	0.1145
Aa	Bb	CC	AaBbCC	0.1167	0.1145
AA	bb	CC	AAbbCC	0.1167	0.1145
Aa	BB	Cc	AaBBCc	0.1167	0.1145
AA	Bb	Cc	AABbCc	0.1167	0.1145
AA	BB	cc	AABBcc	0.1167	0.1145
aa	Bb	CC	aaBbCC	0.1250	0.1225
Aa	bb	CC	AabbCC	0.1250	0.1225
aa	BB	Cc	aaBBCc	0.1250	0.1225
Aa	Bb	Cc	AaBbCc	0.1250	0.1225
AA	bb	Cc	AAbbCc	0.1250	0.1225
Aa	BB	cc	AaBBcc	0.1250	0.1225
AA	Bb	cc	AABbcc	0.1250	0.1225
aa	bb	CC	aabbCC	0.1333	0.1310
aa	Bb	Cc	aaBbCc	0.1333	0.1310
Aa	bb	Cc	AabbCc	0.1333	0.1310
aa	BB	cc	aaBBcc	0.1333	0.1310
Aa	Bb	cc	AaBbcc	0.1333	0.1310
AA	bb	cc	AAbbcc	0.1333	0.1310
aa	bb	Cc	aabbCc	0.1417	0.1402
aa	Bb	cc	aaBbcc	0.1417	0.1402
Aa	bb	cc	Aabbcc	0.1417	0.1402
aa	bb	cc	aabbcc	0.1500	0.1500

The penetrance values are calculated using an expansion of the additive and multiplicative model formulas. The *f0* corresponds to the wild-type genotype of "AABBCC" in which there are no disease associated SNPs present. The penetrance probability for the *f0* is set in this baseline state as 10% to account for phenocopy and other non-genetic factors. The *f1* for the 3 gene model concatenates three analogous *f1* heterozygotes for each SNP, thus the genotype "AaBbCc" represents the *f1* for the 3 locus model and has 3 disease associate alleles present. It follows that the *f2* is represented by the "aabbcc" genotype, having the maximum possible 6 disease associated variants. The formula proposed by Risch and Teng for an additive common disease model is shown in Equation 2.

Equation 2: ***Additive complex disease model developed by Risch and Teng [22]***

$$f2 = 2(f1) - f0$$

The penetrance values corresponding to the genotypes were calculated for the expanded additive model. The set of penetrance probability endpoints were chosen to the *f2* genotype of "aabbcc" corresponding to all 6 alleles being of the disease associated form in all 3 genes. The range of the penetrance endpoints resulted in ten penetrance functions being created. The endpoint penetrance values used the following probabilities for the *f2*: [0.15, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, and 1.0]. These *f2* endpoints were chosen to cover a range of relative risks between the *f0* and *f2* from relative risk 1.5 to relative risk 10 if all 6 alleles are of the disease associated form.

Knowing the probability values f0 and f2 allows for deriving the f1 for the "AaBbCc" heterozygote genotype for each of the ten penetrance function calculations. This is shown in Derivation of Equation 3, and results in Equation 3.

Derivation of Equation 3: ***solving f1 of additive complex disease model using equation 2***

$$f2 = 2(f1) - f0 \quad \text{(Equation 2)}$$

$$\rightarrow f1 = \frac{(f2 + f0)}{2} \text{ (Equation 2 solved for } f1)$$

Equation 3: ***Calculation of f1 penetrance probability for additive complex disease model***

$$f1 = \frac{(f2 + f0)}{2}$$

For example, using the baseline *f0* = "AABBCC" (with its penetrance probability of 0.10) and *f1* = "aabbcc" with probability of 0.15, the *f1* can be calculated and is shown in Sample Calculation for Equation 3.

Sample calculation for Equation3: ***calculation of f1 in additive model given penetrance probabilities (f0 = 0.1) and (f2 = 0.15)***

$$f1 = \frac{(0.15 + 0.1)}{2}$$

$$\rightarrow f1 = \frac{0.25}{2}$$

$$\rightarrow f1 = 0.125$$

The penetrance probability values for the remaining genotypes were calculated by stepwise filling in of the intermediate probabilities based of number of disease associated

alleles present in the genotype. For each model genotype, the penetrance value was calculated by taking the difference in the probabilities from the *f0* and *f1*, subsequently dividing by 6 (the total number of alleles) and multiplying by the number of disease associated alleles present in the genotype as shown in equation 4. The result is an additively linear model of disease penetrance probability with respect to the number of disease associated alleles present in the genotype.

Equation 4: ***General penetrance probability calculation for additive complex disease model***

$$f = [\, f0 + ((f2 - f0)/6) \,] * (\text{number of disease associated alleles})$$

The multiplicative model calculations were based on an expansion of the Risch and Teng complex disease model. Their formula for a multiplicative effect model in a single locus of bi-allelic gene"A" yields three possible states, "AA", "Aa", and "aa". These genotype states and their corresponding associated penetrace probabilities are represented by [*f0, f1*, and *f2*]. Therefore, *f0* = "AA" wild-type (and its corresponding penetrace probability), *f1* = "Aa" heterozygote with 1 disease associated allele, and *f2* = "aa" homozygote with two disease associated alleles all genotypes having a corresponding penetrance probability. The associated penetrance probabilities of the multiplicative model for a single locus model follow the formula in Equation 5.

Equation 5: ***Multiplicative complex disease model proposed by Risch and Teng [22]***

$$f2 = f1^2$$

This formula was the basis for calculating the penetrance functions for a three locus multiplicative model. As with the additive model, the *f0* representing the wild-type genotype "AABBCC" having no disease associated alleles. This *f0* was set to a 0.10 penetrance probability, meaning a 10% disease risk given no disease associated SNPs are present. This is meant to account for environmental and other non-genetic mechanisms of disease contribution. The same *f2* probability endpoints were taken as the additive model [0.15, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, 0.90, and 1.0]. The penetrance values for the *f1* were calculated by taking the square root of the relative risk of the *f2* with respect to the *f0*, and then multiplying by the penetrance probability of the *f0*. For example, in the *f0* = ["AABBCC" with a penetrance probability of 0.1] and the *f2* = ["aabbcc" with a probability of 1.0], then the relative risk (*f2*/*f0*) = 10. The *f1* is calculated by taking the square root of the relative risk and multiplying by the *f0*. This calculation form is shown in the sample calculation for Equation 6.

Equation 6: ***f1 penetrance probability for multiplicative complex disease model***

$$f1 = f0 * \sqrt[2]{Relative\ Risk\left(\frac{f2}{f0}\right)}$$

Sample calculation for Equation 6: ***f1 in multiplicative complex disease model given [f0 = 0.1] and [f2 = 1.0]***

$$f1 = f0 * \sqrt[2]{Relative\ Risk\left(\frac{f2}{f0}\right)}$$

$$\rightarrow f1 = 0.1 * \sqrt[2]{\frac{1.0}{0.1}}$$

$$\rightarrow f1 = 0.1 * \sqrt[2]{10}$$

$$\rightarrow f1 = 0.1 * 3.162$$

$$\rightarrow f1 = 0.3162$$

The penetrance values for the remaining genotypes having 1, 2, 4 or 5 disease associated alleles were calculated by taking the sixth root of the *f2* relative risk and multiplying by the number of disease associated alleles present in the genotype as shown in Equation 7 and Sample Calculation for Equation 7.

Equation 7: ***General penetrance probability calculation for multiplicative complex disease model***

$$\textit{Penetrance probability for multiplicative complex disease model} = (\#disease\ associated\ alleles) * f0 * \sqrt[6]{Relative\ Risk\ \left(\frac{f2}{f0}\right)}$$

Sample calculation for Equation 7: ***multiplicative complex disease model for genotype with 2 disease associated alleles given f0 = 0.1 and f2 = 1.0***

$$\textit{Penetrance probability} = (\#disease\ associated\ alleles) * f0 * \sqrt[6]{Relative\ Risk\ \left(\frac{f2}{f0}\right)}$$

$$\rightarrow \textit{Penetrance probability} = 2 * 0.1 * \sqrt[6]{Relative\ Risk\ \left(\frac{1.0}{0.1}\right)}$$

$$\rightarrow \textit{Penetrance probability} = 2 * 0.1 * \sqrt[6]{10}$$

$$\rightarrow \textit{Penetrance probability} = 0.2 * 1.468$$

$$\rightarrow \textit{Penetrance probability} = 0.2154$$

Figure 2 shows a graph of the penetrance function for additive and multiplicative disease models with an *f0 = 0.10,* and *f2 = 1.0*. The penetrance function for the graph is shown in Penetrance Table 10.

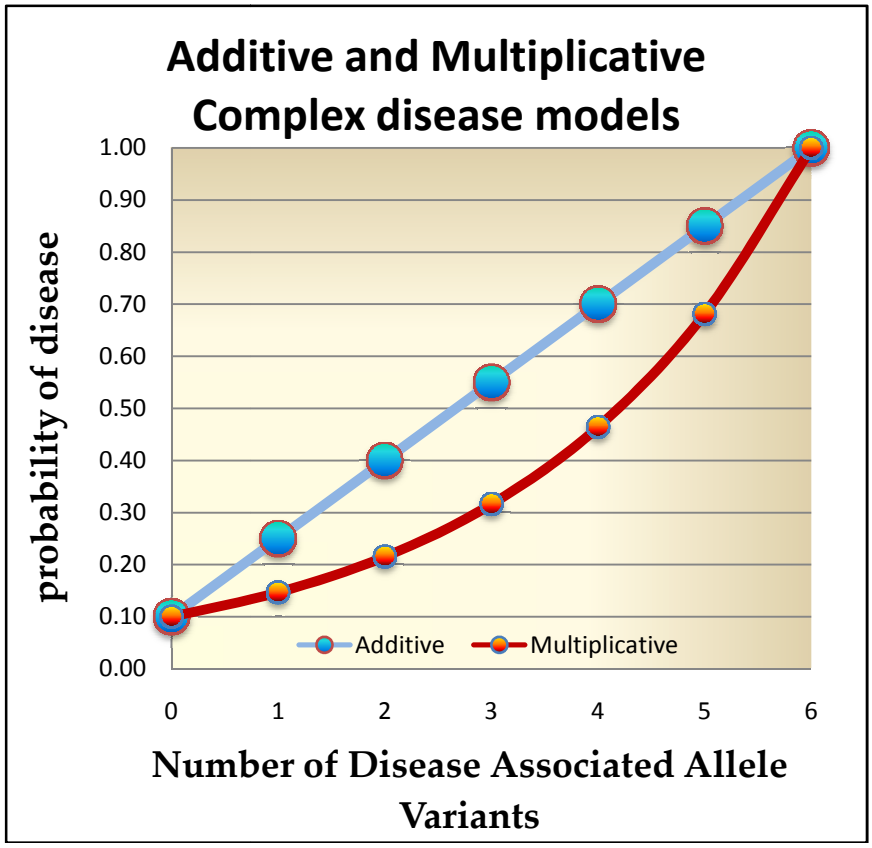


Figure 2: Graph of penetrance functions for Additive and Multiplicative Complex Disease Models.

Relative risk range 2.5 to 10.0

Table 3: Penetrance Table 10 for Relative Risk Range 2.5 – 10.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes with 1 disease allele variant highlighted in yellow
Homozygotes with 2 disease associated variant alleles highlighted in blue

Disease Associated Marker				Disease Risk (Probability)	
A	B	C	Genotype	Additive	Multiplicative
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.2500	0.1468
AA	Bb	CC	AABbCC	0.2500	0.1468
AA	BB	Cc	AABBCc	0.2500	0.1468
aa	BB	CC	aaBBCC	0.4000	0.2154
Aa	Bb	CC	AaBbCC	0.4000	0.2154
AA	bb	CC	AAbbCC	0.4000	0.2154
Aa	BB	Cc	AaBBCc	0.4000	0.2154
AA	Bb	Cc	AABbCc	0.4000	0.2154
AA	BB	cc	AABBcc	0.4000	0.2154
aa	Bb	CC	aaBbCC	0.5500	0.3162
Aa	bb	CC	AabbCC	0.5500	0.3162
aa	BB	Cc	aaBBCc	0.5500	0.3162
Aa	Bb	Cc	AaBbCc	0.5500	0.3162
AA	bb	Cc	AAbbCc	0.5500	0.3162
Aa	BB	cc	AaBBcc	0.5500	0.3162
AA	Bb	cc	AABbcc	0.5500	0.3162
aa	bb	CC	aabbCC	0.7000	0.4642
aa	Bb	Cc	aaBbCc	0.7000	0.4642
Aa	bb	Cc	AabbCc	0.7000	0.4642
aa	BB	cc	aaBBcc	0.7000	0.4642
Aa	Bb	cc	AaBbcc	0.7000	0.4642
AA	bb	cc	AAbbcc	0.7000	0.4642
aa	bb	Cc	aabbCc	0.8500	0.6813
aa	Bb	cc	aaBbcc	0.8500	0.6813
Aa	bb	cc	Aabbcc	0.8500	0.6813
aa	bb	cc	aabbcc	1.0000	1.0000

CHAPTER III

METHODS AND MATERIALS: SOFTWARE, CODING, & BENCHMARKS

GenomeSIM: genome simulation software

In order to generate the simulated genomes for the study, genomeSIM was used [49]. This software package is able to simulate large scale datasets for either population or case/control whole-genome association studies. GenomeSIM is written in ANSI-C++. User specified parameters can include the size of the population, number of genes, number of SNPs per gene, allele frequency ranges of non-causal SNPs, the minor allele frequencies (MAF) for single or multiple causal SNPs, and the penetrance function probabilities for the disease associated alleles. A marked advantage in genomeSIM versus other genome simulation packages is that penetrance functions can be specified to a high degree of detail with multiple disease associated alleles to yield a variety of disease models. Thus, a simple penetrance function with a single locus or more complex penetrance function incorporating multiple markers can be specified to generate dominant, recessive, additive, or multiplicative complex disease models. Other simulation packages were also considered for generation of the datasets. However, these simulators do not easily allow for multiple SNP penetrance functions, limiting their effectiveness for this particular study. Given the nature of complex disease where several disease variants and the interactions between them determine disease status, genomeSIM was chosen. Mainly for its flexibility in implementing common disease models with multiple disease associated alleles.

There are two main modes of genomeSIM operation that allow for either population based or probability (case/control) based creation of datasets. The probability mode was used for the creation of all genome datasets in this work to simulate a case/control study. A 3 locus disease model was created and used for all the study evaluations. Minor allele frequencies (MAF) of the 3 disease markers were given a fixed value of 0.20 (unless otherwise noted); while a range of allele frequencies for the non-causal SNPs were specified from 0.01 to 0.50. The simulator works by assigning disease status from the probabilities of the penetrance function tables. At first, only the three disease associated marker alleles are generated, according to the fixed minor allele frequencies assigned by the user. The user defined penetrance function for the genotype file is then looked up to assign disease or non-disease status based in the probability of disease for each genotype. In its case/control generation mode, genomeSIM proceeds until the desired number of cases and controls are created having only the 3 disease loci. After this step, the remainder of the SNPs for each genome set is generated according to the ranges specified for non-causal SNPs. For this study, all penetrance tables were either completely additive or completely multiplicative in nature. However, genetic heterogeneity can be generated by genomeSIM by using multiple penetrance functions. This is done by assigning the number and percentage of different penetrance table models to use.

GenomeSIM uses a dual chromosome representation for the genotype data. Thus, each individual in the population has two binary chromosomes. The allele frequency values are determined from either the specific allele frequencies for the disease-associated alleles given by the user or sampled from the frequency ranges specified for

the non-associated alleles. The genotype at any locus is represented as 0, 1, or 2 in the genomeSIM genotype output file. This is determined by adding the allele "values" of the two chromosomes at each locus and corresponds to the wild-type homozygote of "AA" = 0+0 = 0, a heterozygote with one disease associated SNP of "Aa" = 0+1 = 1, and a homozygous double mutant "aa" = 1+1 = 2.

Benchmarks: genomeSIM

Given our study design incorporated association testing of many thousands of genotypes at the whole-genome scale, computational tractability is a primary concern. GenomeSIM was housed and run on the Vanderbilt ACCRE cluster using one Intel Opteron processor with 400MB of RAM. According to the benchmarks in the publication of genomeSIM specifications, a 100,000 SNP dataset of 500 cases and 500 controls took ~12 seconds and a 400,000 SNP set was produced in ~50 seconds [49]. Our results show a longer creation time as a 10,000 SNP set of 500 cases and 500 controls took ~15 seconds to create. However, the 12 second benchmark for a 100,000 SNP dataset was clocked on a different processor (Intel Xeon@ 3.06MHz) using 2GB of RAM.

Our study used genotype file arrays for evaluation by either pooling or individual genotyping. Each composite genotype file array created by genomeSIM consisted of 10,000 SNPS for each of the 500 cases and 500 controls (sample size 1,000). Overall, 25,000 of these composite genotype arrays were created. This resulted in 25,000,000 overall individual genotypes with 10,000 SNPs per genotype. GenomeSIM is estimated to have taken ~104.16 hours or 4.34 CPU days to create the genotype files.

Converting genomesim genotype files to haploview format for individual association analysis: genomesim_2_haploview.m

The study was designed to use Haploview as the means of conducting individual genotype association testing. A MATLAB script was written in order to convert the genomeSIM genotype files into a Haploview compatible format. This conversion script, called **"**genomeSIM_2_haploview.m", expands the SNP values from genomeSIM into a Haploview compatible version. The main engine of the conversion script expands the single number allele representation of genomeSIM into a 2 number representation for each locus in the genotype file. This is done by essentially converting the homozygous wild-type "AA" genotype representation of [0] from genomeSIM to [1, 1] for Haploview. Likewise "Aa" and "aA" heterozygotes are converted from [1] in genomeSIM to [1, 2] or [2,1], respectively. An "aa" is converted from [2] in genomeSIM to [2, 2] for its Haploview representation. A Haploview formatted file will contains six accessory information columns in addition to the 20,000 SNP marker columns, which is twice the number of SNPs due to the expansion from single to double binary representation for each locus. In the Haploview compatible file, Column 1 contains the pedigree name. This is a unique identifier for this individual's family. Column 2 is individual ID, here set to sample number 1 to 1000. Columns 3 and 4 are father and mother ID, for trio and pedigree analysis, here they were all set to [0] as unknown. Column 5 is sex, (1=male, 2=female, 0=unknown), here all set to [0]. Column 6 is affection status, (1=control, 2=case) determined from the original genomeSIM disease status column 1. Columns 7 – 20,007 are the marker genotypes. Each marker is represented by two columns (one for

each allele, separated by a space). A [0] in any of the marker genotype position indicates missing data or genotype error.

Individual genotyping with haploview

For individual genotyping by association testing, Haploview analysis software was used [50]. Haploview is written in JAVA and conducts single point association analysis using the standard Pearson chi-squared test. The chi-square derived p-values from the allele frequencies in case versus control individuals are the output of the association testing. Further, permutation testing with the individual genotyping association test results can correct for multiple testing bias. Additional features allow linkage disequilibrium (LD) measures such as D', r^2, and Log of Odds (LOD) to be calculated as well as haplotype block analysis, haplotype population frequency estimation [50]. For the purposes of this study, the single point association testing feature of Haploview served as a standard for individual genotyping for comparison of the pooled association testing by our pooled genotyping analysis tool (sm_PDA). Although a GUI interface for Haploview is the default operating mode, the program was run in command line mode form the windows command line in order to speed up analysis and conduct individual genotyping association analysis in batch mode. Memory use was doubled from the default 512MB to 1GB of RAM and resulted in a notable decrease in processing time. Given that each dataset consisted of 1,000 cases and controls at 10,000 SNPs, and that there were 100 datasets each for as many as 15 parameter points, this modification became significant in terms of time savings. Haploview took ~2 minutes to process each genotype file for each individual genotype.

As a check for Haploview association testing for individual genotyping, our pooled analysis tool (sm_PDA) was used with a sampling error of 0% which approximates the association pooling statistic from Equation 1 to that of a chi-square distribution with 1 degree of freedom. The association test statistic for pooled cases versus controls has two sources of variance in its denominator, measurement error (V_1) and sampling error (V_2) of Equation 1.

Equation 1: ***Pooled genotyping association testing statistic***

$$z^2 = \frac{(p_1 - p_2)^2}{V_1 + V_2}$$

V_1 is the variance due to measurement error and arises from the genotyping platform in the form of genotyping error. This measurement error is present in both individual and pooled genotyping. V_2 in the pooled association testing is variance due to sampling error essentially form errors in allele frequency estimation by the genotyping platform and/or from unequal amounts of DNA comprising the pools. Sampling error is not present in individual genotyping. If the V_2 sampling error is theoretically reduced to 0%, then the only source of variance in the pooling statistic becomes measurement error (V_1). Consequently, the pooling association testing statistic in Equation 1 approximates to a chi-square distribution with one degree of freedom. This is what is used for individual genotyping by Haploview. This served as a reciprocal check between Haploview and sm_PDA to insure that genotype files were being created, processed, and analyzed properly. Additionally, as our modified version of PDA ran in ~2.3 *seconds* compared to Haploview taking ~2 *minutes*, a potentially useful by-product could be to use sm_PDA

for individual genotyping analysis. Test value statistics using Haploview and sm_PDA on the same genotype files showed nearly identical chi-square derived p-values.

Converting genomesim genotype files to pda format for pooled genotyping analysis: genomesim_2_pda.m

In our study, the **Pooled DNA Analyzer (PDA)** program was used to conduct pooled association testing [51]. The original PDA code was obtained from the developers. We extensively modified the PDA by adding batch mode capability, re-organizing the input and output results interfaces, and optimizing the source code resulting in a nearly *50-fold faster performance*. Our modified version of PDA was termed **sm_PDA**. In order to convert a genomeSIM output genotype files to PDA compatible files, a MATLAB script termed "genomeSIM_2_PDA.m" was created. This script scores the allele frequencies for each marker in the control and case groups. The output file is formatted as an sm_PDA compatible file. For example, a genomeSIM genotype file is a [1,000 row by 10,001] column matrix where each row represents a case or control individual and columns represent the 10,000 SNPs plus 1 column for disease status. The first column of to 10,001 total shows the disease state of the individual (0 for case, 1 for control) and columns 2 through 10,001 are the locus allele "values" of the binary chromosome (0, 1, or 2). By convention, in a genomeSIM genome file, a locus with the genotype "AA" is scored as "0", "Aa" or "aA" is "1", and "aa" is "2". genomeSIM_2_PDA.m converts the genomeSIM data into an allele frequency file suitable for sm_PDA. This is done by scoring each genotype at each SNP and summing the frequencies of the normal "A" and disease associated "a" alleles. GenomeSIM_2_PDA.m scores the allele frequencies for the cases and controls of the

genotype dataset and lists them sequentially in the output file. The result "PoolAF" file is a 20,002 row by 5 column matrix of allele frequencies along with other information. The first column is disease status, 1 for controls and 2 for cases. The second column lists the marker "name", in our case the marker number. The third column is the total number of cases or controls in the dataset. The forth column is the major allele frequency. The fifth column is the minor allele frequency (MAF). The major and minor allele frequencies are scored by expanding each locus "score" form genomeSIM file into an allele frequency matrix. An "AA" homozygous wild-type with no disease allele variants is represented by a [0] in the genomeSIM datafile. GenomeSIM_2_PDA.m will add 2 to the "wild-type" signifying the presence of two wild-type alleles at this SNP locus. A heterozygous "Aa" or "aA" represented as [1] in the genomeSIM file will add 1 to the "A" array element and 1 to the "a" element of the allele frequency matrix, signifying one normal allele and one disease associated variant allele are present at that locus. It follows that a homozygous double mutant "aa" which is [2] in the genomeSIM file will add 2 to the "a" element of the array at its corresponding locus, thus representing the presence of 2 disease alleles at that locus. This is processing step connects the genomeSIM genotype file output by transforming it into PDA formatted allele frequency file for subsequent pooled association analysis.

Benchmarks: genomesim_2_pda

Using the PROFILER function of MATLAB, the total time required to convert a genomeSIM genotype file to a PDA compatible format was evaluated. A genomeSIM file containing 10,000 SNPs for 500 case and 500 control individuals took 19.5 seconds

to convert. The genomeSIM_2_PDA conversion script has three main sections. It first loads the genomeSIM genotype text file into active memory as matrix. Second, the genotype matrix is passed to the scoring function with a function call to the calculation function. Third, the scored pooled allele frequencies are written as a delimited text file using the MATLAB DLMWRITE function. Looking at the time breakdowns for each function within the automated conversion script, 70% of the total time (13.7 of 19.5 total seconds) is spent opening the genomeSIM genotype text file. The second most time consuming part of the script is the allele scoring function itself, which takes 4.1 of the total 19.5 seconds, corresponding to 21% of the total time. Thirdly, the writing of the allele frequency results to a tab delimited file using DLMWRITE takes 1.7 seconds, 8.8% of the total time. Overall, the 25,000 genotype files analyzed for the study are estimated to have taken 135.4 CPU hours or 5.6 CPU days to pre-process from the genomeSIM genotype files to PDA compatible input files. Note that this pre-processing step is significantly longer than the actual pooled association testing analysis by sm_PDA, which takes ~2 - 2.5 seconds per file, estimated at 16.56 days 0.69 CPU days overall for the 25,000 genotype files tested.

Creation of sm_PDA by modifications to PDA (the pooled dna analyzer) for pooled association analysis:

In order to conduct pooled genotyping associating analysis a modified version of the Pooled DNA Analyzer (PDA) program was created [51]. The PDA is a MATLAB implemented suite of modules which allows single point association testing of pooled DNA using the pooled association testing statistic (Equation 1). Additional functionality allows for chromosome-wide multipoint association tests based on p-value combinations

using a sliding-window concept. Although PDA is designed primarily for pooled associational analysis, individual genotyping can also be conducted by either single point or multipoint test by setting the sample error to 0%, effectively decomposing the pooling association test statistic into a chi-square test with 1 degree of freedom, which is what is used for individual genotyping. This feature of PDA was used as a double-check for the individual genotyping conducted by Haploview, as well as a check to confirm that the creation and processing of the pooled allele frequency files was performed correctly by correlating the Haploview individual genotyping with the PDA "individual" genotyping results. Additionally, after the original version of PDA was modified into sm_PDA, individual genotyping with sm_PDa could be run at a 50-fold reduced time.

The original version of the Pooled DNA Analyzer (PDA) implements a GUI where the user directs a path to a folder containing 3 necessary files to run a pooled association analysis. The first of these 3 files is a column vector with the designated names or number of each marker, termed "SNPname". The second file used by PDA is "IndPI" and is the basis for signal correction and normalization for the association testing. The "IndPI" (Individual Peak Intensity) file is a four column matrix which contains the SNP name, its map location (optional), and relative signal intensities for homozygous and heterozygous individuals at each SNP. Its purpose is to normalize the signal intensities of the allele variants at each genotyping feature of the genotyping platform. The purpose of this file is to reduce sampling error in pooling by correcting biased signal intensities or platform artifacts which are a major source of allele frequency estimate error. For our study, all peak intensities of the "IndPI" file were defaulted to 1. In practice, databases exist in which known heterozygote and homozygote intensities of

the SNPs are cataloged and serve as a reference for normalization and correction [7, 52, 53]. The third and most directly evident file needed for PDA is the "PoolAF" (Pooled Allele Frequencies) file which contains major and minor allele frequencies of the case and control groups. The "PoolAF" file consists of the disease or group identifier (1=case, 0=control), the SNP name (matching the "SNPname" file), the number of individuals in each pooled group (number of cases or number of controls), and the allele frequencies for the major and minor alleles at each locus. These three PDA processing files need to be located in the same folder, which can be specified by the user from the PDA GUI.

The PDA GUI has eight edit boxes and over 20 check boxes which are used to set the parameters for pooled association testing, shown in Figure 3.

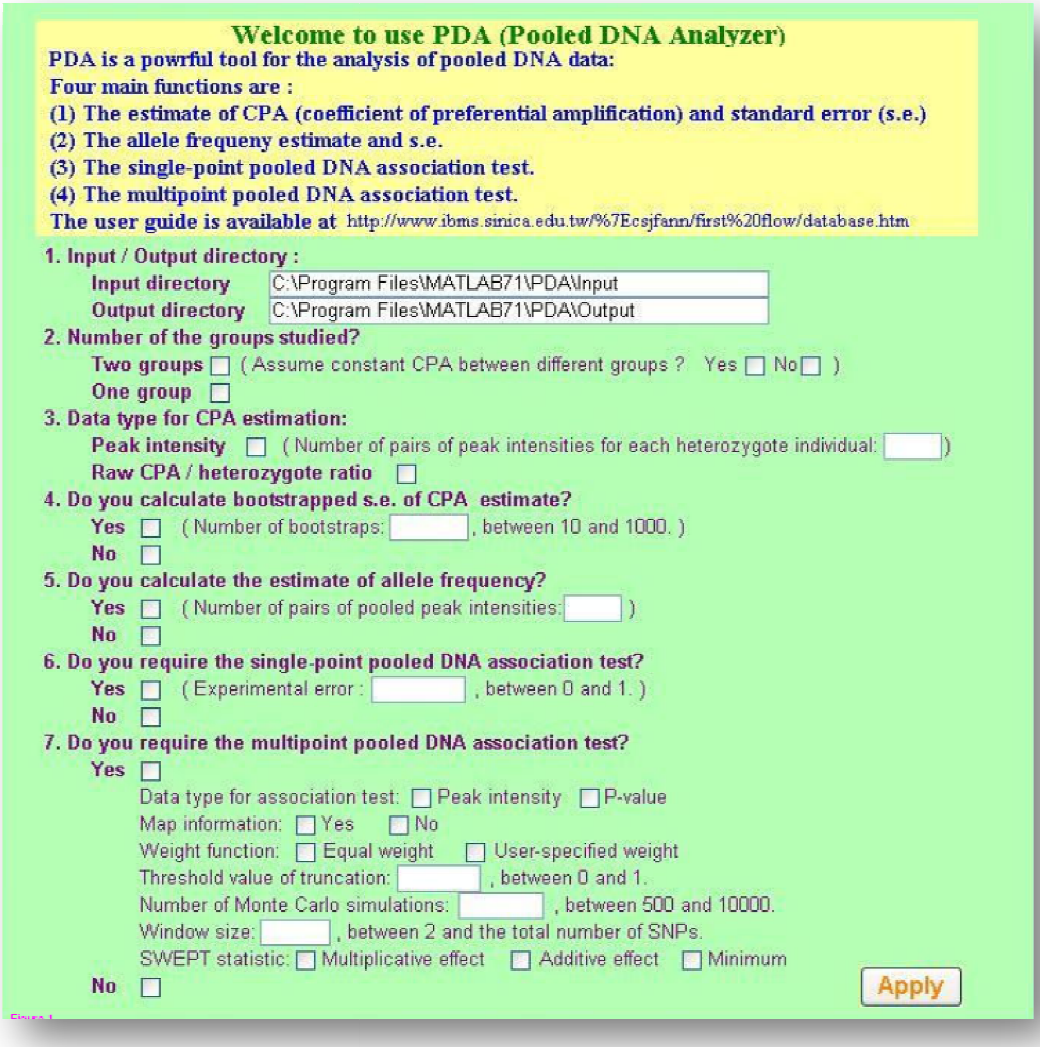


Figure 3: Pooled DNA Analyzer GUI screen

There are seven parameter fields in the PDA GUI which are combinations of edit and check boxes. The parameter fields are:

1) Edit boxes for the input and output path directories,

2) Check boxes for the number of groups in the study, (one group or two) and check boxes to assume a constant Coefficient of Preferential Amplification (CPA) between the groups if a two group study.

For this study, a two group test for cases and controls with constant CPA assumed.

3) Check boxes for the method of peak signal normalization. Either using the Coefficient of Preferential Amplification (CPA) method [51] or the raw peak intensity.

For this study, raw peak intensity was used with both the homozygote and heterozygote intensities defaulted to 1.

4) Check box to calculate the standard error and the number of bootstraps to use.

For this study, 500 bootstraps were used.

5) Check box to calculate the estimate of allele frequency.

Checked to "yes" in the study.

6) Check box to perform the single point association test and the sample error to incorporate.

This is the main functionality to perform pooled association analysis. Sample error was specified from 0% to 5%.

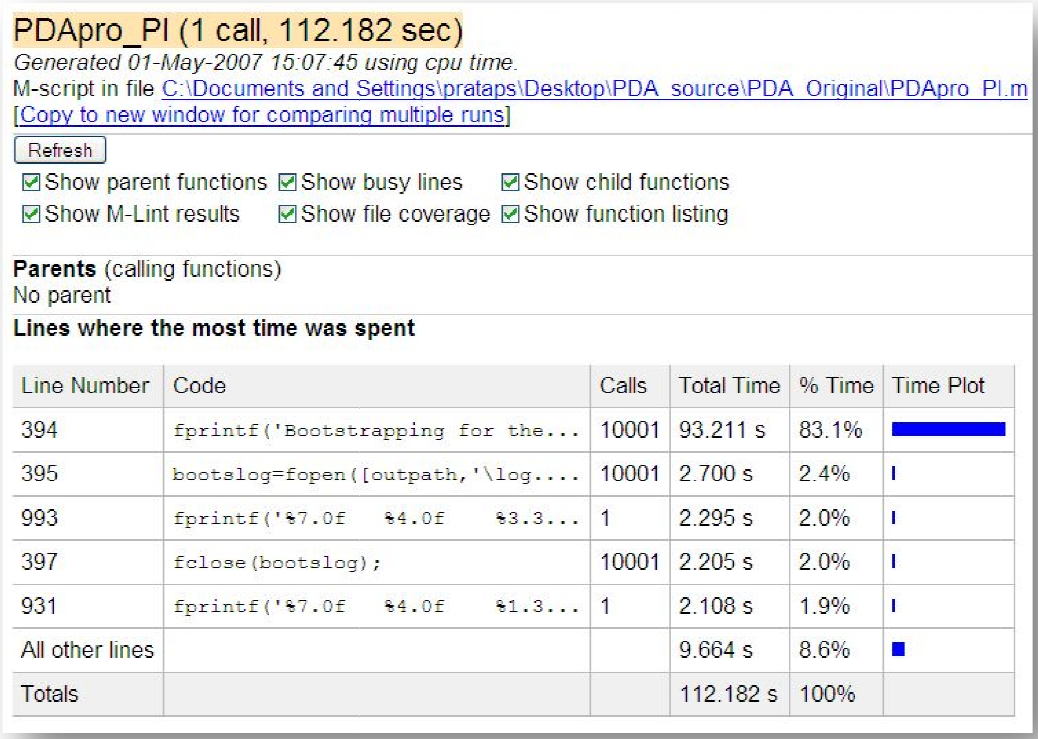
7) Multiple check boxes and edit fields to perform multi-point association testing.

- Whether to use p-values from single point association or peak intensities.
- Whether map information is present.

- The weight functions for peak intensities.
- A threshold value for truncation.
- The number of Monte Carlo simulations to use for permutation testing.
- The window size to use for multipoint testing.
- The disease model effect to use for the SWEPT multipoint test: multiplicative, additive, or minimal.

In order to evaluate the time profile characteristics of the Pooled DNA Analyzer (PDA), the MATLAB PROFILER was used. The PROFILE function of MATLAB analyzes the amount of time spent on the execution of functions within a MATLAB script. Using the PROFILE VIEWER mode in MATLAB, the results are output as an HTML file. The PROFILER can output the time in seconds as well as percent of total time spent in each line or function of a script. It also has a feature which details the clocked times between parent and child functions (additional functions called from the parent). Additionally, the number of times a function is called are also listed.

The original PDA code was run over a dozen times using the PROFILER function in order to clock its speed and the time distributions of the various blocks of code. The results of a representative PDA run time profile are shown in Figure 4.



PDApro_PI (1 call, 112.182 sec)

Generated 01-May-2007 15:07:45 using cpu time.

M-script in file C:\Documents and Settings\prataps\Desktop\PDA_source\PDA_Original\PDApro_PI.m

[Copy to new window for comparing multiple runs]

Refresh

☑ Show parent functions ☑ Show busy lines ☑ Show child functions

☑ Show M-Lint results ☑ Show file coverage ☑ Show function listing

Parents (calling functions)

No parent

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
394	`fprintf('Bootstrapping for the...`	10001	93.211 s	83.1%	
395	`bootslog=fopen([outpath,'\log....`	10001	2.700 s	2.4%	
993	`fprintf('%7.0f %4.0f %3.3...`	1	2.295 s	2.0%	
397	`fclose(bootslog);`	10001	2.205 s	2.0%	
931	`fprintf('%7.0f %4.0f %1.3...`	1	2.108 s	1.9%	
All other lines			9.664 s	8.6%	
Totals			112.182 s	100%	

Figure 4: MATLAB PROFILER for PDA.

results for time to run original PDA code @ 112 seconds and percent of time spent in function calls:

Results from the time analysis show that the original version of PDA took 110-120 seconds to analyze a pooled genotype file. Given that the project would need to process over 25,000 genotype files, using the original version of PDA would take more than 50,000 minutes, or *34.722 days of CPU time* to calculate association analysis for the pooled genotypes. In addition to the slow processing time, the original PDA did not have batch capability and was not automated by any definition of the word. For this reason, a speed enhanced version of PDA (sm_PDA) was created which would have a batch operation mode as well as reduce the time to process a pooled genotype file for association testing.

SM_PDA: added batch processing and automation to the original PDA

One of the main reasons for modifying the original version of PDA into sm_PDA was to add batch mode functionality, allowing for association analysis of the datasets in an automated fashion. As the project involved association analysis and genotyping of tens of thousands of genotype files by pooled allelotyping, this automation and batch mode functionality were critical. In order to conduct a pooled analysis with the original version of PDA, the pooled allele frequency files ("PoolAF" files) generated form genomeSIM_2_PDA.m would need to be moved into the PDA input directory folder *one at a time*. Then the user would hit the "APPLY" button in the PDA GUI to perform association analysis. Once the pooled analysis was completed, the results file would have to be moved to another folder to avoid being overwritten by the next analysis. This process would need to be repeated for each genotype file and was thus not feasible for the

more than 25,000,000 genotypes which were analyzed in the pooled genotyping phase of the study.

Batch modifications to the original PDA code allowed for processing the thousands of genotype files in each phase of the pooled analysis to be analyzed without manually setting up and moving each file for processing by PDA. This allowed for a major improvement in automation of the program as well as reduced time to process the association testing. The basic procedure for batch mode operations was to enclose the single and multipoint testing processes within a three nested FOR_LOOP structure. In order to use the FOR_LOOP syntax, the pooled allele frequency file input name parameter in the original PDA was modified to accept user specified file names (instead of only "PoolAF") using the SPRINTF and EVAL functions of MATLAB. For example, a round of genotype files created in genomeSIM would have 100 replicates for a given disease model at a given relative risk. Using the genomeSIM_2_PDA.m script, each of these genotype files would be converted to a Pooled Allele Frequency file with appended alphanumeric tags; "PoolAF_α_β_γ_δ". The first filename modifier (α) was the sub study being done such as relative risk, genotype error rate, or population size. The second filename modifier (β) is the disease model of the genotype simulation, "add" for additive or "multi" for multiplicative models. The third appended tag (γ) is replicate number as 100 replicates were done for each data point. In general, multiple alphanumeric tags were appended to the "PoolAF" files during their creation from the original genomeSIM genotype data. Upon pooled genotyping analysis with sm_PDA, the SPRINTF and EVAL functions of MATLAB were used in order to process the "PoolAF_α_β_γ" files into application memory in an automated batch mode. Using this

modification, the "PoolAF_α_β_γ" file could be referenced by the correlated FOR_LOOP iteration parameters. The overall result was an automated process which did not require moving individual "PoolAF" pooled allele frequency files into the input folder one at a time as in the original PDA. Further, the pooled association testing result files did not need to be moved out of the results folder in order to avoid being overwritten by the subsequent results file. This was among the most significant modification made in the creation of sm_PDA and represented the added functionality of an automated program with true batch mode capability, something that the original PDA was not able to do.

SM_PDA: logical operator modifications

In the MATLAB code for the PDA, the GUI implemented check boxes and edit boxes are coded into logical conditions that set the parameters for association analysis. In the original PDA code, the logical checks are written in a way that evaluates the *entire* logical expression and then returns the result. If the logical expressions are long, than evaluation of the entire expression may not be necessary, and could become time consuming and inefficient. For example, any "false" element of a logical expression connected with "and" will result in a "false" evaluation of the entire expression. If the "false" element occurs during evaluation of the expression, checking the remaining elements is not necessary in this case. The many hundreds of logical conditions form the original PDA code were re-coded using "short circuit" logical connectors in order to eliminate unnecessary checks. For example, the first logical evaluation line of the original PDA processing script is shown in Box 1.

Box 1: original version of pda logical expression syntax for check box evaluation:

```
if check1==1 & check2==1 & check5==1 & check7==1 & check9==1 &
   check11==1 & check12==1 & check14==1 & check16==1;
```

The "&" is the logical operator "and" in MATLAB and the line is a logical expression with eight conditional checks all connected by the logical "and" operator. For this statement to be true, all of the conditionals must be true. If any single conditional is false, then the entire statement is evaluated as false. With this statement syntax as it is written in the original code, each element of the entire expression is evaluated first, and then the logical value of true of false is assigned to the statement as a whole. This is a potentially wasteful exercise if, for example, the first "check" is false. This renders the entire expression as false, yet all of the remaining checks will be evaluated. The short circuit "and", coded in MATLAB as "&&", was used in these logical statements in order to speed up the code. In this syntax, each element of the expression is evaluated and, as soon as one element is false, the expression returns false without any further element evaluations. Likewise, the logical "or" was replaced with "short circuit or" which will return "true" if any of the elements are true and will not evaluate the expression further. Given this change, all of the logical expressions should be evaluated with the fewest possible number of elements being evaluated. Although logical expressions are relatively quickly evaluated in MATLAB, given the fact that there are nearly 100 potential logical evaluations for each genotype file tested, and 25,000 association tests files are evaluated

in this study, the time savings are potentially significant. Overall reduction with implementation of the logical short circuit operators, in part, contributed to a speed up in the association testing calculations of at least 4 fold. The original PDA was clocked at 9.6 seconds for that section of the program code to run, while the entire modified sm_PDA code ran in 2.3.

SM_PDA: result output file modifications

Another modification to the original PDA program made in order to optimize and automate the process was to change the formatting and writing characteristics of the association testing result files. In the original version of PDA, the results were written to a log file in "real-time". Specifically, the association statistics of each marker were written to the log file as soon as the association processing was done for each marker. This mixing of association test processing with input/output transitions for writing to the results file is extremely inefficient. Significant time could be and was saved by performing all of the processing of the association testing first, storing the final results in a matrix or application memory, and only then writing to an output file. Additionally, input/output transitions which updated the progress of PDA to the console after the processing of every SNP, every bootstrap, and each function module were eliminated. The original version of PDA used FPRINTF statements to write the output results to a log file. The FPRINTF function writes formatted data and (in the original version of PDA) writes each line of association testing results as they are calculated. This was changed in sm_PDA by using the DLMWRITE function which is faster, albeit less versatile, than FPRINTF. Sm_PDA saves the pooled association testing results as a matrix during the

calculations and writes them as a delimited text file only when all the association testing is completed. According to the MATLAB designers, DMLWRITE is a faster function than FPRINTF, but the major time savings with this modification are a result of eliminating the writing of each line of results to both the log file and the console as they are calculated from the association testing.

Sm_PDA outputs tab delimited text file containing the marker number, the pooled association testing chi-square value, and the chi-square correlated p-value. Results of the PDA modifications into sm_PDA are a nearly 50-fold reduction in pooled association analysis. These time analysis comparison results are shown in detailed in BENCHMARKS: OriginalPDA and speed modified sm_PDA.

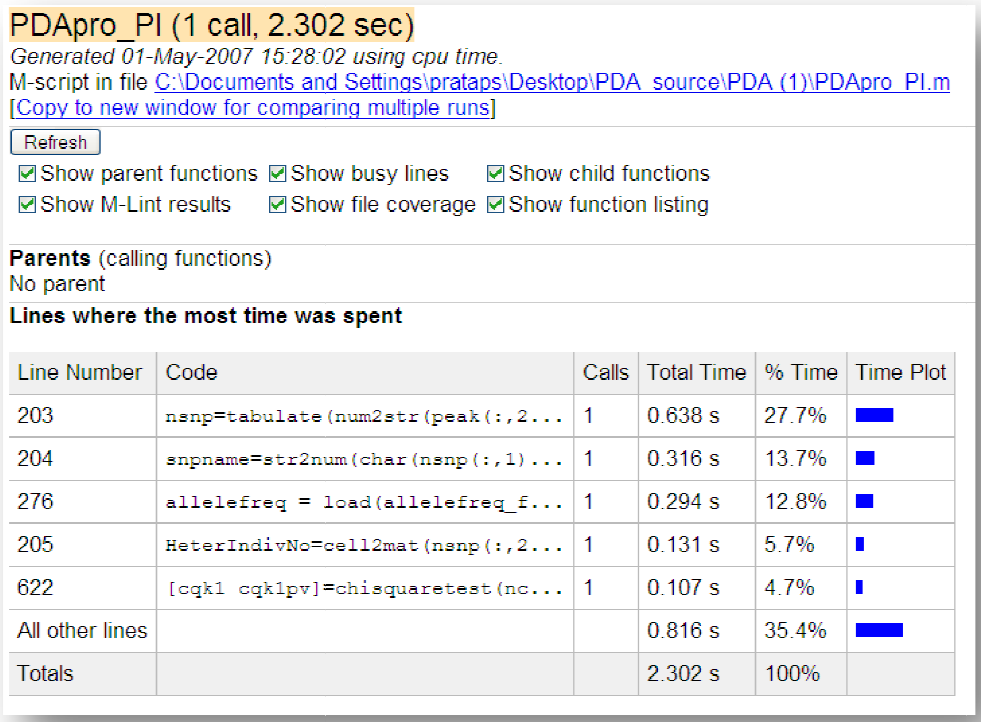
Benchmarks: original PDA and speed modified SM_PDA

The original version of PDA was evaluated for its time to process a pooled allele frequency file using the MATLAB PROFILER. The total time for evaluating one genome file was >110 seconds. Using the PROFILE function, it was shown that the majority of CPU time was used for real time progress messages sent to the MATLAB console window of the standard output (i.e. monitor) and creation of the results logfile as shown in Figure 4.

In total, over 90% of the CPU time was used for two tasks related to outputting results. 83% of the total CPU time (93 of 112 seconds) was used for progress updates during the Bootstrapping stage to determine the chi-square p-values of association testing. The second most time consuming task was opening, writing to, and closing the

log file output of the association analysis. This results log process used 4.9 seconds or 4.4 % of the total time to open and close the log file 10,000 times (once for each marker), and 4.4 seconds or 3.9% of the total time writing the associating results to the log file. Adding up the log file open/close and writing results and progress to the console and output log showed that over 91.6% of the total time by the original PDA code was used there. Note that none of the association testing was done in these steps, only the output writing. Thus, the time involved in actual association testing using the Pooled DNA Analyzer (PDA) software is not the most time consuming step at all, surprisingly. All remaining tasks, *including the association analysis itself* were performed in the remaining 9.6 seconds or 8.4% of the total 112 seconds. Calculating for the original PDA version, 25,000 genotype files times 112 seconds each yields 777 CPU hours or 32.4 days of total CPU time that would have been required for pooled analysis had sm_PDA not been created and used.

A MATLAB PROFILER time analysis of the modified version, **sm_PDA** is shown in Figure 5. The figure is representative of over a dozen runs of sm_PDA with time clocking from the PROFILER.



PDApro_PI (1 call, 2.302 sec)

Generated 01-May-2007 15:28:02 using cpu time.

M-script in file C:\Documents and Settings\prataps\Desktop\PDA_source\PDA (1)\PDApro_PI.m

[Copy to new window for comparing multiple runs]

Refresh

☑ Show parent functions ☑ Show busy lines ☑ Show child functions

☑ Show M-Lint results ☑ Show file coverage ☑ Show function listing

Parents (calling functions)

No parent

Lines where the most time was spent

Line Number	Code	Calls	Total Time	% Time	Time Plot
203	`nsnp=tabulate(num2str(peak(:,2...`	1	0.638 s	27.7%	
204	`snpname=str2num(char(nsnp(:,1)...`	1	0.316 s	13.7%	
276	`allelefreq = load(allelefreq_f...`	1	0.294 s	12.8%	
205	`HeterIndivNo=cell2mat(nsnp(:,2...`	1	0.131 s	5.7%	
622	`[cqk1 cqk1pv]=chisquaretest(nc...`	1	0.107 s	4.7%	
All other lines			0.816 s	35.4%	
Totals			2.302 s	100%	

Figure 5: CPU time for sm_PDA.

MATLAB PROFILER results for time to run the modified sm_PDA code @ 2.3 seconds compared to 112 seconds for the original PDA (Figure 4).

Overall time for sm_PDA is *2.3 seconds* compared to *112 seconds* for the original PDA version. The total CPU time for processing the 25,000 genotype files was estimated to be 16.66 CPU hours or 0.69 CPU days to process. Thus, with modified sm_PDA, the total time to process the 25,000 genotype files is decreased by a factor of 46 fold compared to the original PDA. The overall time reduction for processing the pooled genotype files was reduced from 32.4 CPU days for the original PDA to 0.69 CPU days for sm_PDA. More importantly than the time savings from speed optimizations, sm_PDA added a batch mode functionality which was critical in allowing the 25,000 pooled genotype files to be analyzed in an automated process.

Parsing the results from pooled and individual association analysis: pda_2_pval.m and haploview_2_pval.m

Once the sm_PDA processed the pooled genotype files, 100 association analysis test result replicates per data point were produced. In order to combine the overall results from the pooled association tests, a MATLAB script termed "PDA_2_pval.m" was created. The function of this script is to combine the results of the repetitions of the sm_PDA association testing and determine the average the p-values for the causal SNPs from the association analysis. Additionally, the standard deviations were calculated using the MATLAB STANDARD function. The standard deviation can be used along with the number of genotype file repetitions in order to derive the confidence intervals.

The PDA_2_pval.m script functions by first loading the 100 sm_PDA pooled association test results files for each repetition set. The results are loaded into a matrix and then sorted by chi-square p-value. The p-values of the 3 disease associated SNPs are averaged over the one hundred genotype repetitions as well as their respective average

ranks. Additionally, the standard deviations of the causal SNP p-values over the replicates are calculated. These results are output into two files, one listing the average p-values, and the second listing the standard deviations of the p-values. A similar MATLAB script, "Haploview_2_pval.m", parses the association testing results from Haploview individual genotyping. The Haploview_2_pval.m script parses the chi-square derived p-values, and averages them for the 100 replicates per data point. The average rank and standard deviations of p-values over the 100 replicates for each data point are also calculated.

CHAPTER IV

RESULTS

Results: sample size

The sample size in the association study and the ability to identify the causal variant alleles given the sample size was evaluated for pooled versus individual genotyping. GeneomSIM was used in its probability mode to create a case / control populations from 200 to 1,000 individuals. GenomeSIM assigns disease status to each individual in the population as their genotype is generated. In the population mode of genomeSIM, this disease status assignment will continue until the specified numbers of cases and controls are generated. The genomeSIM input parameter files were constructed to output datasets with varying populations from 100 to 500 disease cases (in increments of 100) and an equal number of controls. This yielded 5 populations of 200 to 1000 (incremented by 200) comprised of equal numbers of case and controls for comparison by pooled allelotyping and individual genotyping. These populations were generated for three levels relative risk; 1.5, 2.0, and 2.5 using Penetrance Tables 4, 7, and 10, respectively. Finally, additive and multiplicative models were both used to generate the populations.

The genotype files were generated on the VANDERBILT Advanced Computing Center for Research & Education (ACCRE) cluster using an Intel Opteron processor with 400MB of RAM. For this study of the effect of sample size on individual genotyping versus pooling, 3,000 total genotype files were generated; 100 replicates for 5 sample size populations at 3 relative risks levels and 2 complex disease models.

The genomeSIM genotype files were converted by the genomeSIM_2_PDA.m pre-processing script for pooled genotyping with sm_PDA. The pooled genotype files were then processed by sm_PDA with a 2% sample error rate. The results were parsed by PDA_2_pval.m and chi-square p-values of the causal SNPs were obtained. The averaged p-values for the 100 replicated of each data point were obtained.

In parallel the genomeSIM genotype files were also converted to Haploview compatible files for individual genotyping with genomeSIM_2_Haploview.m. The individual genotyping results were parsed from the Haploview association testing result files with Haploview_2_pval.m.

The results of the pooled allelotyping with 2% sampling error were compared to the individual genotyping. Significance level was set to p-value $<5e^{-2}$. The average chi-square p-value for 100 replicate genotype files versus the number of cases and controls for additive and multiplicative disease models at three relative risks (1.5, 2.0, and 2.5) were graphed. These results are shown in Figure 6.

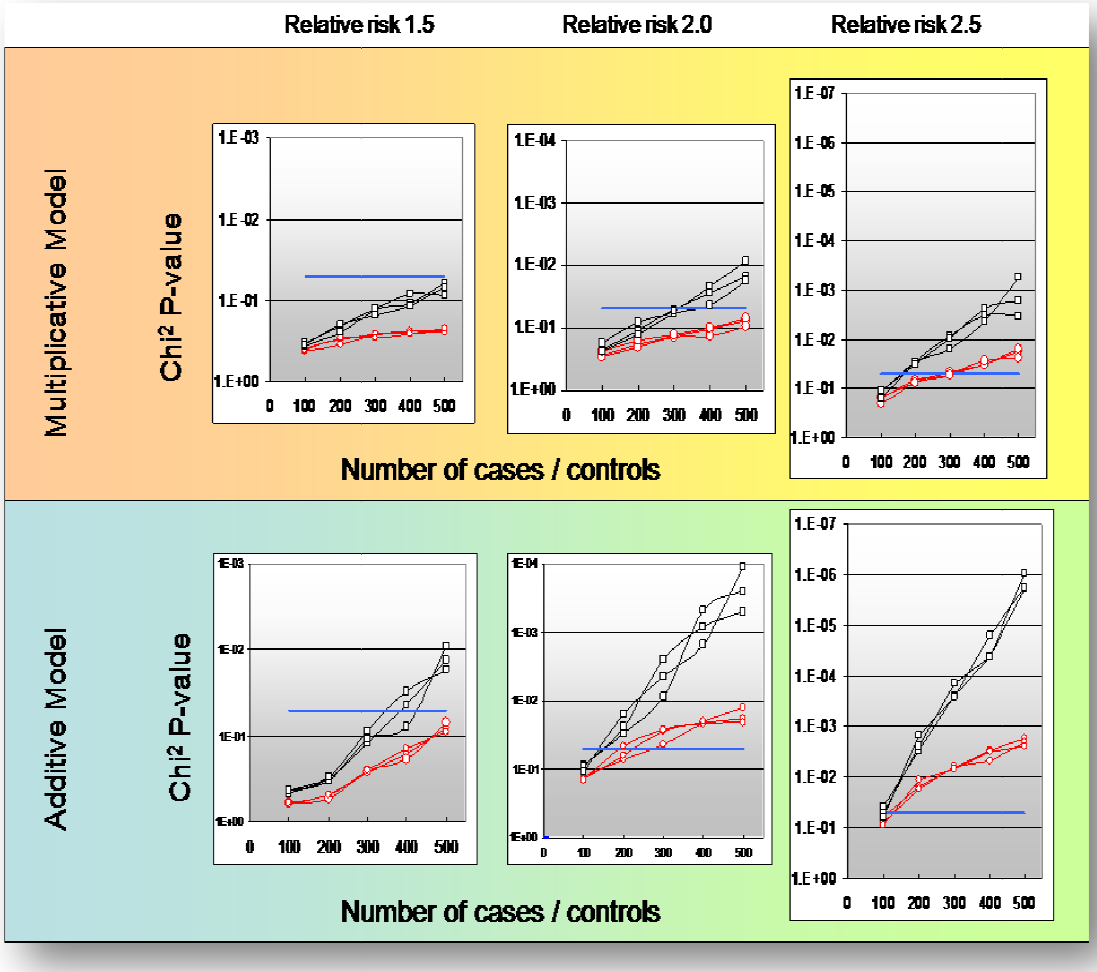


Figure 6: Effect of Smaple size on pooling versus individual genotyping

Individual genotyping: white squares with black lines:
Pooled genotyping with 2% sampling error: red circles with red line.
Cutoff threshold (p-value = $5e^{-2}$): blue horizontal line

Data points represent average p-value of 100 simulations

The results show that for the **multiplicative model** with a relative risk of 1.5, neither pooled (Figure 6: red lines with circles) nor individual genotyping (Figure 6: black lines with squares) was able to identify any of the 3 causal SNPs with a p-value above the threshold of $5e^{-2}$ for all population sizes tested (Figure 6: upper left panel). At relative risk of 2.0, the three causal SNPs were identified with a population of 400 cases and 400 controls, while the pooled genotyping for the multiplicative model at relative risk 2.0 did not identify any of the causal SNPs, even up to 500 cases and 500 controls (Figure 6: upper middle panel). At a relative risk of 2.5, the individual association analysis crossed the $5e^{-2}$ p-value threshold identifying the 3 causal SNPs with 200 cases and 200 controls, while pooled genotyping identified 1 of the causal SNPs with 200 cases and controls and all 3 SNPs at 300 cases and 300 controls (Figure 6: upper right panel).

For the **additive model**, 2 of the 3 disease associated SNPs were identified at relative risk 1.5 with individual genotyping of 400 cases and 400 controls, and all 3 markers were identified with 500 cases and 500 controls. Results for pooled genotyping additive model at relative risk 1.5 show that none of the populations tested yielded p-values above threshold (Figure 6: lower left panel). For relative risk 2.0, individual genotyping with 200 cases was sufficient to identify all 3 disease SNPs while pooled genotyping identified 1 of the causal SNPs with 200 cases and all 3 causal SNPs with 300 cases and 300 controls (Figure 6: lower middle panel). For relative risk 2.5, individual genotyping was able to identify 1 causal with 100 cases and all 3 with 200 cases while pooling identified all 3 causal SNPs with 200 individuals. (Figure 6: lower left panel)

Results: relative risk

In order to evaluate the resolution of pooling and individual genotyping over varying relative risks, genomeSIM was used to create genotypes with a range of relative risks. The relative ranges varied from 1.07 for a single disease associated to 1.5 for the corresponding 6 variant genotypes in the least penetrant model (Penetrance Table 1) ; up to 2.5 for a single disease associated and 10.0 for the corresponding 6 variant genotypes variant (Penetrance Table 10) (see Penetrance Tables 1-10). The relative risks were tabulated in the penetrance tables for both additive and multiplicative models. A total of 20 penetrance tables were constructed; 10 for the additive complex disease model and 10 for the multiplicative model. The baseline disease probability was set to 10% for all penetrance functions. This 10% baseline represents the probability of disease given no disease associated alleles. This 10% baseline accounts for non-genetic factors such as environmental conditions (phenocopy) which contribute to disease status and reflects a more realistic complex disease model as environmental factors significantly contribute to complex disease. Additionally, the base rate of 10% includes disease associated alleles which may not be present on the SNP chip and other non-genetic entities which contribute to disease progression.

Relative risk is defined as the ratio of the probability of disease given the case/disease genotype versus the probability of disease given the control/normal group genotype. For example, given a baseline of 10% risk of disease in the control group, if the probability of disease in the case group is 20%, then the resulting relative risk is 2.0, [(0.20 / 0.10)]. Therefore, the case group is twice as likely to have the disease compared to the control group. The penetrance functions in genomeSIM were constructed using the

Risch and Teng complex disease model formulas as a foundation [48] (see Methods and Materials: Complex Disease Model Construction). The "AABBCC" genotype represents the non-disease control for the three gene model with 6 wild-type, non-disease associated alleles. The probability of disease for the control genotype is set to 10% to allow for phenocopy. In the complex disease model formula, this genotype and its associated penetrance probability is represented as *f0*. The penetrance values for the model genotypes were calculated using an expansion of the additive and multiplicative model formulas (Equations 2 and 5). The *f0* corresponds to the wild-type genotype of "AABBCC" in which there are no disease associated alleles present in the diploid genotype and its resulting probability. The penetrance probability for the *f0* is set in this baseline state as 0.10 to account for phenocopy and other non-genetic factors. The *f1* for the expanded 3 marker model concatenates three analogous *f1* for each locus "Aa", "Bb', and "Cc". Thus, the genotype "AaBbCc" represents the *f1* for the 3 locus model. It follows that the *f2* is represented by the "aabbcc" genotype to represent 6 disease associated alleles; the 3 markers with 2 disease associated alleles each. Expanding the formula proposed by Risch and Teng for complex disease models resulted in an increase from 3 possible genetic states (3^1 for a single gene model) in the original model to 27 possible genetic sates (3^3) with a 3 locus model.

The penetrance functions were calculated by setting the probability of the *f0* to 0.10 and varying the probability of the *f2;* from [0.15, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.0]. The expanded model formulas were then used to determine the penetrance probabilities for the intermittent genotypes including the *f1* as well and the remaining 24 genotype states (see Penetrance Tables 1-10). By convention, the penetrance functions

were labeled by the relative risk of 1 disease associated allele being present versus the *f0* with no disease associated alleles. For example, if the *f0* = ["AABBCC" with penetrance probability of 0.10], the *f2* = ["aabbcc" with a penetrance probability of 1.0], then the penetrance probability of 1 disease associated allele is 0.25, calculated by Equation 4.

Equation 4: ***General penetrance probability calculation for additive complex disease model***

$$f = [\,f0 + \left(\frac{f2 - f0}{6}\right)] * (\text{number of disease associated alleles})$$

The relative risk of the single disease associated allele relative to the *f0* is (0.25 / 0.1) = 2.5. Thus, the penetrance function for the additive disease model having *f0* = 0.1 and *f2* = 1.0 has a relative risk of 2.5 for a single disease associated allele being present. This penetrance function was labeled as *"penetrance table with relative risk 2.5"* for reference. The penetrance function of the corresponding multiplicative model having *f0* = 0.1 and *f2* = 1.0 was also designated as relative risk 2.5. However, a single disease associated allele in this multiplicative model will not result in a relative risk of 2.5 as does the additive model with the same *f0* and *f2* parameters. The beginning and end points are equivalent in terms of penetrance probability. This naming convention allows for comparison between the additive and multiplicative models.

For the additive and multiplicative models of complex disease, GenomeSIM was used to create 100 replicates for each of the 10 relative risk points [1.08, 1.16, 1.33, 1.5, 1.66, 1.83, 2.0, 2.16, 2.33, and 2.5]. A total of 2,000 genotype files were created and subsequently analyzed. Each genotype file was of 10,000 SNPs for 500 cases and 500 controls. The genomeSIM files were converted to PDA and HAPLOVIEW compatible

genotype files by GenomeSIM_2_PDA.m or GenomeSIM_2_HAPLOVIEW.m scripts, respectively. The converted genotype files were processed by sm_PDA with a 2% sampling error rate for the pooled genotyping and HAPLOVIEW, for individual genotyping. The results were parsed by PDA_2_PVAL.m or Haploview_2_PVAL.m. The average chi square p-value of the 100 individual or pooled genotypes was plotted versus the relative risk. These results are shown in Figures 7 and 8.

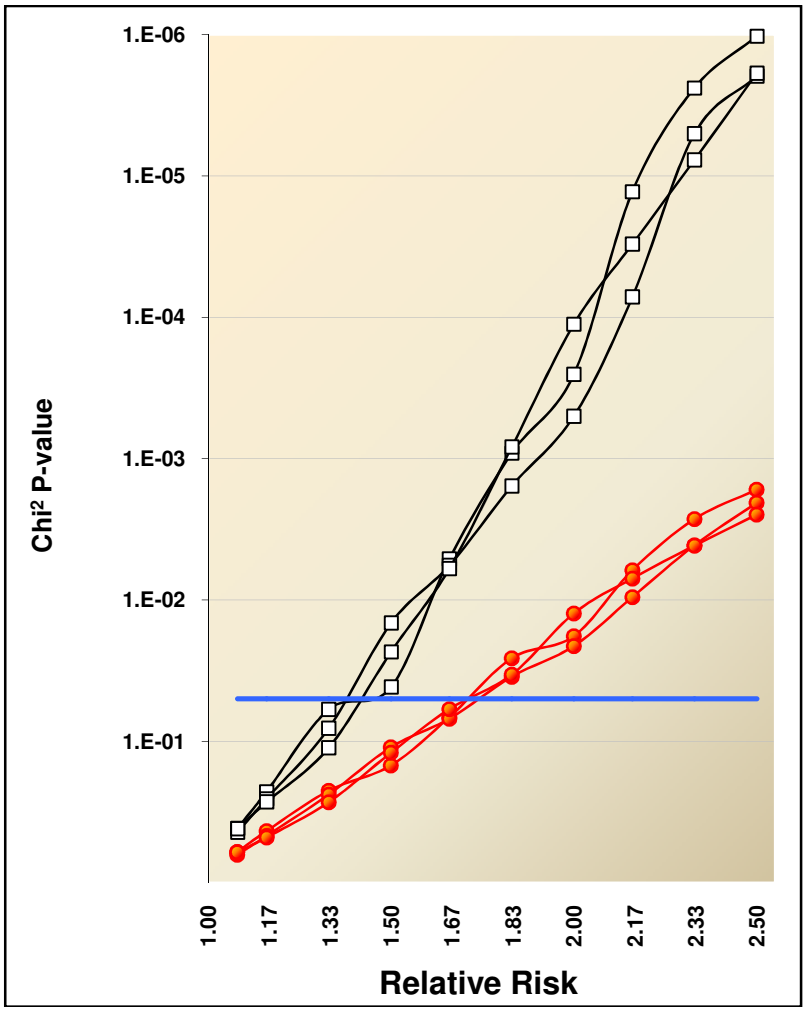


Figure 7: Individual versus Pooled genotyping at varying relative risk ranges

Individual genotyping: white squares with black lines:
Pooled genotyping with 2% sampling error: red circles with red line.
Cutoff threshold (p-value = $5e^{-2}$): blue horizontal line

Data points represent average p-value of 100 simulations
Additive Complex Disease Model

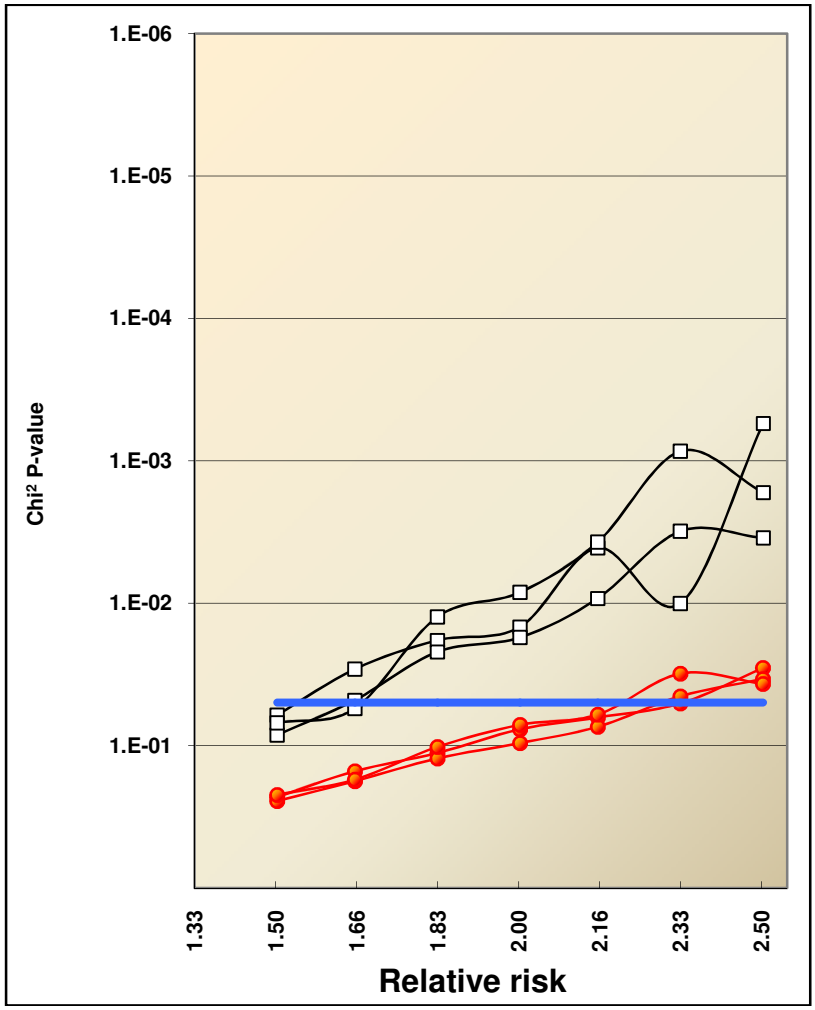


Figure 8: Individual versus Pooled genotyping for a Multiplicative Model of complex disease

Individual genotyping: white squares with black lines:
Pooled genotyping with 2% sampling error: red circles with red line.
Cutoff threshold (p-value = $5e^{-2}$): blue horizontal line

Data points represent average p-value of 100 simulations

For the **additive model**, individual genotyping was able to detect all 3 causal SNPs above threshold ($<5e^{-2}$) at a minimum relative risk of 1.5. The results for pooling with a 2% sampling error rate showed that the 3 disease associated alleles were detected above threshold at a minimum relative risk of 1.83 (Figure 7). For **multiplicative model**, individual genotyping was able to detect 2 of the 3 causal SNPs at a relative risk of 1.66, and all 3 causal alleles at a relative risk of 1.83. For pooled genotyping with a multiplicative model, 2 of the 3 causal SNPs were detected at a relative risk of 2.33, and all 3 disease variant markers were detected at relative risk 2.5 (Figure 8). The gap in detection resolution between pooling and individual genotyping (with respect to relative risk) was wider in the multiplicative model compared to the additive model. The detection level of the 3 disease associated alleles in the additive model was relative risk 1.5 for individual genotyping and 1.83 for pooling. This yielded a minimum relative risk differential of 0.33 relative risk units for the pooling to detect the 3 causal SNPs. This differential was wider in the multiplicative model. Individual genotyping resolved the 3 causal SNPs at a relative risk of 1.83, while pooling resolved at a relative risk of 2.5, a difference of 0.67 relative risk units. Thus the gap of the relative risk at which individual genotyping and pooled genotyping are able to resolve the disease associated alleles is twice as large for the multiplicative model (0.67 relative risk) versus the additive model (0.33 relative risk). The conclusion from the data suggests that the multiplicative model of complex disease requires a higher level of relative risk conferred by the disease associated alleles for detection compared to the additive model. This was true for both individual and pooled genotyping, but more pronounced in pooling. These results further

suggest that complex diseases with an additive characteristic of disease penetrance are more amenable to a pooling approach.

Results: genotyping error

In order to evaluate the effects of genotype error on pooled and individual genotyping, GenomeSIM was used to create files with 10,000 SNPs and 1,000 samples in the population (500 cases and 500 controls). Genotype error was incorporated at levels of ranging from 0% to 10%, with 1% increments. In terms of the simulations, genotype error is a parameter which can be specified during the creation of the genotypes and is derived by inserting an "unknown" call at a marker position to simulate a genotyping error from the genotyping platform. The marker position at which the "unknown" call is located is based upon a uniform random distribution and the user defined percentage of genotype error. As each locus is assigned a value during the creation of the genotype, a random number is generated from a uniform distribution and checked against the percentage of genotype error specified in the genomeSIM parameter input file. The "unknown" flag representing a genotype error is inserted based upon this process and inserted if the sampled number is below the threshold set for the genotype error.

GenomeSIM was used to create 100 replicates of genotype files with 11 levels of genotype error ranging from 0% to 10% (with 1% increments). This yielded 1,100 genotype files in total for the genotype error evaluation. The additive model penetrance function in Penetrance Table 7 was used for the creation of the simulated genotypes. The relative risk of a single disease variant allele from Penetrance Table 7 is 2.0; meaning that the presence of a single disease associated allele raises the probability of disease from

0.10 to 0.20. Additional mutations will contribute in a purely additive manner, thus 2 mutations have a relative risk of 3.0; 3 mutations will have a relative risk of 4.0; and 6 mutations result in a relative risk of 7.0. This penetrance function was chosen based on the results of the relative risk evaluation in which both individual and pooled genotyping were able to detect all 3 disease associated alleles with a chi-square p-value $<5e^{-2}$ at relative risk of 2.0 (Figure 7). This relative risk level was also chosen because it is among the first levels of relative risk at which both individual and pooled genotyping results were in excess of the p-value threshold of $5e^{-2}$. Therefore, this level of relative risk would be among the most vulnerable to having it p-values fall below the significance threshold. The results from the genotype error study are shown in Figure 9.

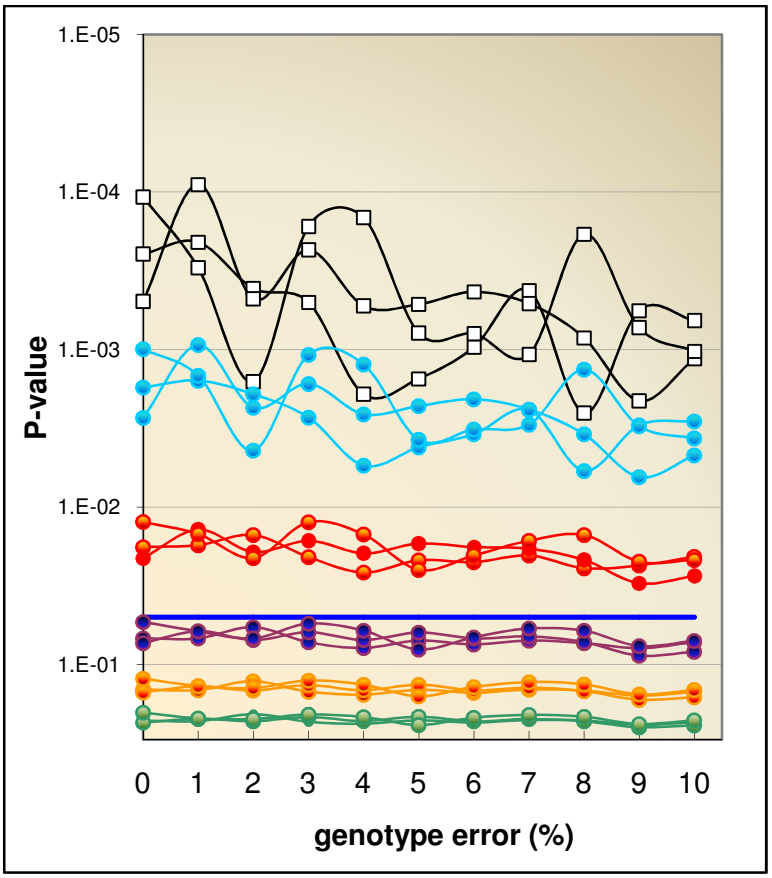


Figure 9: Effect of Genotype Error on Individual and Pooled Genotyping:

White squares with black lines — Individual genotyping

Blue circles and lines — **Pooled @ 1% Sampling Error**
Red circles and lines — **Pooled @ 2% Sampling Error**
Purple circles and lines — **Pooled @ 3% Sampling Error**
Orange circles and lines — **Pooled @ 4% Sampling Error**
Green circles and lines — **Pooled @ 5% Sampling Error**

Datapoints are average p-values of 100 simulations

Individual genotyping at 0% genotype error had p-values of [$1.5e^{-4}$, $2.4e^{-4}$, and $5.0e^{4}$] for the 3 associated SNPs (averaged over the 100 repetitions). At 10% genotyping error, the p-values are [$6.6e^{-4}$, $1.1e^{-3}$, and $1.0e^{-3}$]. Although there is a noticeable decline in the significance of the p-values, it is less than an order of magnitude and well above threshold cutoff of $<5e^{-2}$. This was also the case for all levels of genotype error tested in the intermittent values of 1%-9% genotype error. Overall, genotype error up to and including 10% lowered the significance of the individual genotyping p-values by less than one order of magnitude compared to perfect genotyping with 0% error.

Pooled genotyping was tested at levels of genotyping error from 0% to 10%, with 1% increments. Additionally, sampling errors from 0% to 5% (with 1% increments) were used for pooled genotyping in order to determine the effects of genotyping error in combination with varying levels of sampling error. Pooled genotyping with 0% sampling error represents a theoretical baseline control as the association testing statistic reduces to a chi-square test when the variance form sampling error is 0% (Equation 1). As such, pooled genotyping with a 0% sampling error will have nearly identical statistical results as individual genotyping. This was the case in the genotyping error evaluation as pooling with 0% sampling error had the same p-values as individual genotyping.

At all levels of pooled genotyping tested (with sampling error 1% to 5%), the decreased significance of p-values was much less than one order of magnitude (Figure 9). In fact, the level of sampling error in pooled genotyping was much more of a factor in overall effect on the p-values and dominated the effect of genotyping error at all levels tested. Pooled genotyping with a 0% genotyping error and a 1% sampling error rate had less significant p-values than individual genotyping with a 10% genotyping error rate.

Further, even a 1% percent increase of sampling error in pooled genotyping lowered the significance of p-values more than a 10% increase in genotyping error. For example, an increase of sampling error from 1% to 2% is more detrimental to the results of pooled genotyping than an increase on genotyping error from 0% to 10% within the same sampling error percent.

Results: pooling specific errors: allele frequency measurement error and sample mixing error

In order to evaluate the effect of sample mixing and allele frequency measurement error on pooled genotyping, GenomeSIM was used to create genotype files having 10,000 SNPs for 500 case and 500 controls at varying relative risk levels. An additive complex disease model was employed for the sampling error analysis study. Using Penetrance Tables 1-10, additive common disease models having each of the 10 relative risk range levels were used. These ranges were from relative risk 1.08 to 2.5 for a single disease allele present to 1.5 to 10.0 for all 6 alleles being disease variants. For each of the 10 relative risk levels, 100 replicates were generated to yield 1,000 total genotype files for this phase of the study. GenomeSIM_2_PDA.m and genomeSIM_2_Haploview.m were used to convert the files into sm_PDA and HAPLOVIEW format, respectively. Individual genotyping was performed by HAPLOVIEW in the command line mode with its default settings except for a memory increase from 500MB to 1GB. Pooled genotyping was done by sm_PDA and the effects of sampling error rates from 0% to 5% were tested.

Errors in allele frequency estimation form pooling result in the sample error variances. They are inherent to pooling and result from DNA quantitation methods,

uneven mixing to form the pools, and the sensitivity (minimum resolvable allelic frequency difference) of the genotyping platform. This error can be reduced reliably to <5% [37, 54], and as little as 1% in the absence of experimental bias [37, 55]. The Pooled DNA Analyzer code allows the user to specify the level of sampling error as a parameter in one of the edit boxes of the PDA graphical user interface. The sampling error rate is incorporated into the association testing statistic by PDA and the variance due to sampling error for pooling (V_2 of Equation 1) is derived directly from the sampling error. The results are shown in Figure 10.

Equation 1: *Pooled association testing statistic.*
the variance due to sampling error is represented by "V_2" and highlighted

$$z^2 = \frac{(p_1 - p_2)^2}{V_1 + V_2}$$

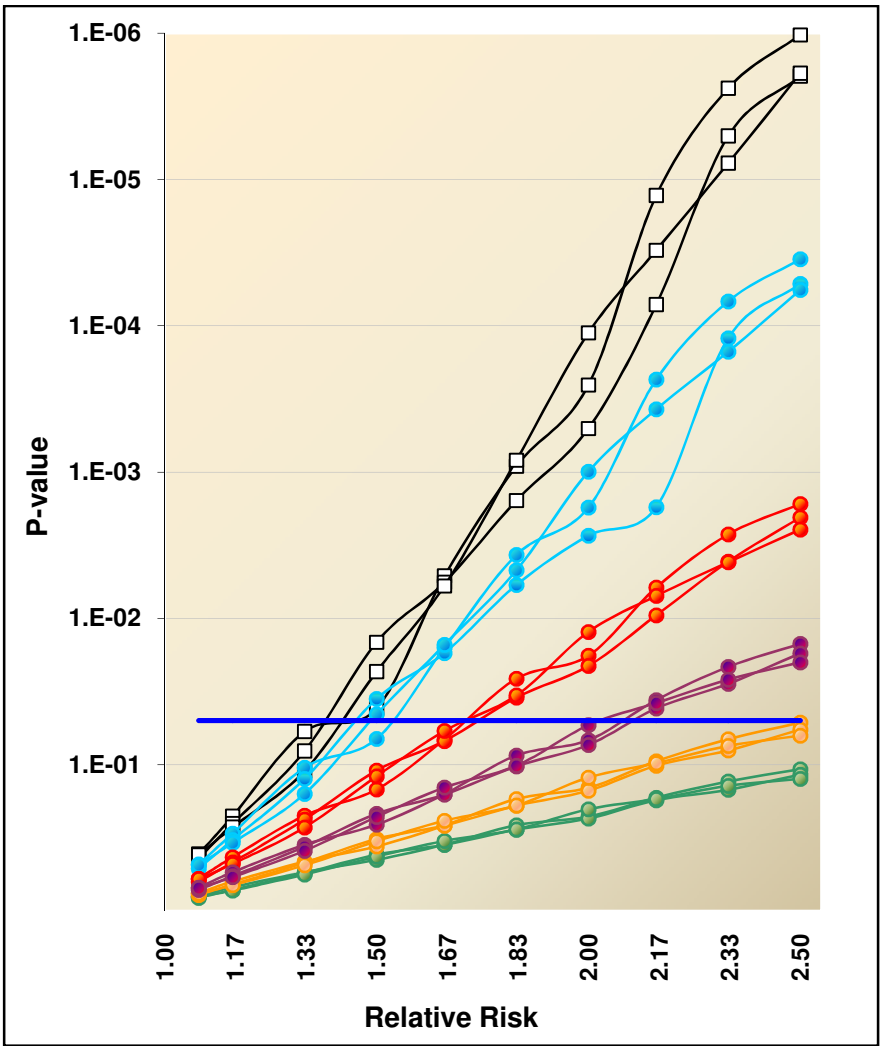


Figure10: Effect of Sampling Error (1% - 5%) on pooled genotyping versus individual genotyping:

White squares with black lines	Individual genotyping
Blue circles and lines	**Pooled @ 1% Sampling Error**
Red circles and lines	**Pooled @ 2% Sampling Error**
Purple circles and lines	**Pooled @ 3% Sampling Error**
Orange circles and lines	**Pooled @ 4% Sampling Error**
Green circles and lines	**Pooled @ 5% Sampling Error**

Datapoints are average p-values of 100 simulations
Additive Complex Disease Model

Individual genotyping resolved all three disease associated SNPs at relative risk of 1.5. Pooled genotyping with a 1% rate of sampling error resolved 2 of the 3 disease SNPs at relative risk 1.5, while all three SNPs were detected at the next relative risk level of 1.67. A 2% sampling error rate increased the minimum resolvable relative risk one more level to 1.83. When the sampling error was 3%, minimum resolvable relative risk increased by two levels to 2.17. At sampling error rates of 4% and 5%, none of the disease SNPs had p-values surpassing the threshold cutoff of $<5e^{-2}$ for any of the relative risk values tested.

Overall, the effect of sampling error on pooled genotyping was to dramatically reduce the significance of the disease associate variant p-values at all levels of relative risk tested. As tested, the effect of sampling error increases the minimum resolvable relative risk at which the association testing p-values of the disease associated SNPs exceeds the threshold p-value. In cases of 4% and 5% sample error, the p-values of the causal SNPs were weakened by 4 to 5 orders of magnitude compared to individual genotyping or pooling with 0% sampling error. This reduction in p-value significance results in none of the disease associated SNPs having p-values more significant than our threshold of $<5e^{-2}$.

CHAPTER V

DISCUSSION

Allele frequency parameter for complex disease models

For this study, a minor allele frequency (MAF) of 0.20 was chosen based on the average MAF in the Caucasian (CEU), Han Chinese/Japanese (CHB+JPT), and African Yoruba (YRI) HAPMAP populations [26, 35, 41]. The Illumina HumanHAP550 genotyping platform shows that the mean minor allele frequencies (MAF) on a genome-wide scale from the total allele frequency distributions are 0.23, 0.21 and 0.22 for the CEU, CHB+JPT and YRI populations, respectively [56]. Median MAFs are 0.23 for CEU, 0.20 for CHB+JPT, and 0.21 for the YRI populations. See Figure 11.

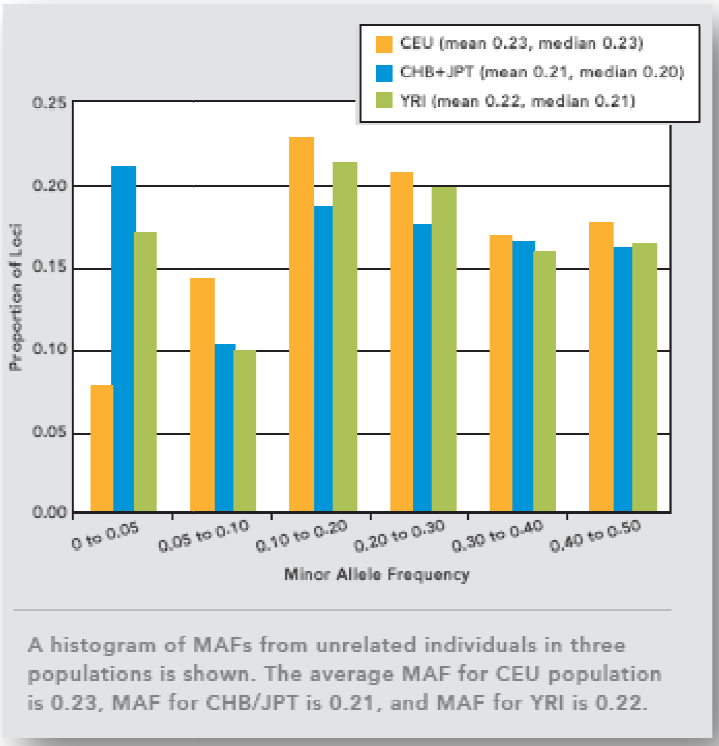


Figure 11: Distribution of Minor Allele Frequencies (MAF) in human HAPMAP populations.

Determined by Illumina HumanHAP550 genotyping platform [56]

As additional evidence for this decision of setting the MAF of the disease associated variants at 0.20, several highly publicized genome-wide association studies on type 2 diabetes (T2D) are cited [6, 12-14, 47]. These studies have lead to a very promising advance in the understanding of T2D by finding or expanding the number of disease associated variants to 10 for T2D. For example, Sladeck and Rocheleua et. al. [12] report that 7 of the 8 T2D associated SNPs in their findings had MAFs from 0.23 to 0.35. The odds ratios of 7 of the 8 the associated SNPs are tightly clustered and range from 1.15 to 1.26 for the heterozygous and 1.36 to 1.53 for the homozygous state. These results are shown in Table 4.

Table 4: type-2 diabetes GWA by Sladek et. al [12]:

Column listings: 1) disease associated SNP name, 2) chromosome, 3) position, 4) risk allele, 5) major allele, 6) MAF of cases, 7) MAF of controls, 8) odds ratios of heterozygotes (het), and 9) odds ratios of homozygotes (hom) [12]

SNP	Chromosome	Position (nucleotides)	Risk allele	Major allele	MAF (case)	MAF (ctrl)	Odds ratio (het)	Odds ratio (hom)
rs7903146	10	114,748,339	T	C	0.406	0.293	1.65 ± 0.19	2.77 ± 0.50
rs13266634	8	118,253,964	C	C	0.254	0.301	1.18 ± 0.25	1.53 ± 0.31
rs1111875	10	94,452,862	G	G	0.358	0.402	1.19 ± 0.19	1.44 ± 0.24
rs7923837	10	94,471,897	G	G	0.335	0.377	1.22 ± 0.21	1.45 ± 0.25
rs7480010	11	42,203,294	G	A	0.336	0.301	1.14 ± 0.13	1.40 ± 0.25
rs3740878	11	44,214,378	A	A	0.240	0.272	1.26 ± 0.29	1.46 ± 0.33
rs11037909	11	44,212,190	T	T	0.240	0.271	1.27 ± 0.30	1.47 ± 0.33
rs1113132	11	44,209,979	C	C	0.237	0.267	1.15 ± 0.27	1.36 ± 0.31

The picture emerges of many risk variants, each having a small and similar magnitude of effect on disease probability. Thus, there is strong support that T2D is a complex disease with an additive type allelic spectrum. The top ranked SNP had a higher odds ratio of 1.65 for the heterozygous state and 2.77 for the homozygous state, as well as a higher MAF of 0.40. The higher MAF and odds ratio of the top ranking SNP likely contributed to the fact that it was the first of T2D associated variants to be found [57]. TCF7L2 was previously associated with T2D by Grant et. al. in 2006. This finding itself evolved from previous work that suggested chromosome 10 (among others) housed a T2D associated region [58]. Importantly, this TCF7L2 association was confirmed in genotypically distinct and diverse populations [59-61], this was a critical finding for a true disease associated variant in humans and not just a spurious genotyping anomaly of a subset. The current group of T2D associated SNPs was found by genome-wide association studies. These studies were able to confirm the TCF7L2 variant found previously. In addition, several more T2D associated SNPs were also discovered. All of the novel SNPs have lower relative risk or odds ratios than TCF7L2 [12-14, 47] and thus lower effect size towards disease contribution. This progress is truly exciting in terms of the pace at which understanding the nature of complex diseases such as T2D is accelerating. The story continues to progress and the methods continue to evolve. The diabetes risk variant question is far from answered; the fact that the list of confirmed genetic variants associated with T2D has increased by an order of magnitude in only the first round of genome-wide association studies suggests that there may be even more associated variants to be found.

The results of the diabetes studies serve as a validation of model parameter choices in as far as they are realistic and seen in the real world. The MAF was fixed at 0.20 for all genotype files and was the first parameter of the models that was decided, based on the diabetes data and HAPMAP data.

Relative risk ranges for complex disease models

Once the MAF of 0.20 was chosen, the range of relative risks was the next parameter to consider. One of the first genome-wide studies to be conducted was by Klein et. al. on age-related macular degeneration (AMD) in 2005 [62]. The conclusion was that 2 variants within the complement factor H (CFH) region were associated with AMD. The study found that the odds ratios of the two associated SNPs were 4.6 to 4.7 for heterozygotes and 7.5 for homozygotes. Concurrently, a linkage screen and resequencing on a candidate region in chromosome 1q32 by Haines et. al. co-confirmed that alleles within the CFH region were associated with AMD [63]. The odds ratios for these variants ranged from 2.45 to 5.5. These results point to a relatively large effect size from a few markers and as such were used as a soft upper ceiling for the model construction parameters in the evaluation. Complex disease is characterized as having many disease markers each with a modest affect, and as such ranges of odds ratios well below the AMD study results were also evaluated. In contrast to the AMD allelic spectra of 2 disease variants with large odds ratios (2.5 to 7.5); the type 2 diabetes profile has 8-10 associated variants each with much smaller odds ratio; 1.19 is the smallest heterozygote odds ration and 2.77 is the largest homozygous odds ratio (Table 4) [12-14, 47, 57, 59-61]. The T2D data was used as a guide to model additive complex disease.

The smallest odds ratio used in this study is 1.07 (multiplicative model) or 1.08 (additive model) representing heterozygotes in 1 of markers with 1 disease associated variant present ["AaBBCC", "AABbCC", or "AABBCc"] (Penetrance Table 1). The maximum relative risk on the study was 10.0 (multiplicative and additive models) corresponding to homozygotes in all 3 markers ["aabbcc"] (Penetrance Table 10). The characteristics of the penetrance tables are better described by the ranges of relative risks that they cover. For example Penetrance table 1 has a relative risk of 1.07 and 1.08 for the multiplicative and additive models with a single disease variant, and increases the relative risks in proportion to the number of variants present in the genotypes up to relative risk 1.5 for 6 disease alleles. The model equations are based on the complex disease formulas derived from the Risch and Teng models for complex disease [40] and listed in Equation 4 for the additive model and Equation 7 for the multiplicative model.

Equation 4: ***General penetrance probability calculation for additive complex disease model***

Penetrance probability for additive complex disease model

$$(f) = [\, f0 + \left(\frac{f2 - f0}{6}\right)] * (\text{number of disease associated alleles})$$

Equation 7: ***General penetrance probability calculation for multiplicative complex disease model***

Penetrance probability for multiplicative complex disease model

$$(f) = (\#\text{disease associated alleles}) * f0 * \sqrt[6]{\text{Relative Risk}\left(\frac{f2}{f0}\right)}$$

The result is a distribution of relative risk levels which conform to the Risch and Teng model definition, while being expanded to a 3 locus model with 27 combinatoric

genotype possibilities. This results in the range of relative risks for the least penetrant model being 1.07 for 1 variant to 1.5 for 6 variants in the smallest effect models (Penetrance Table 1) and the most penetrant model having relative risks ranging from 2.5 for 1 variant present to relative risk 10.0 with 6 disease associated variants (Penetrance Table 10). Penetrance Tables 1-10 are listed in the Appendix.

Population sample sizes parameters for complex disease models

One the parameter choices of MAF at 0.20 and the relative risk ranges were set in the complex disease parameters, the range of sample sizes was the next parameter to define. A study by Zou and Zhao looked at the effect of genotyping and sampling errors on the sample size required to achieve statistical significance in association studies [39]. This evaluation used the Risch and Teng complex disease models as a basis for genotype and penetrance value constructions [48]. The genotype state of wild-type "AA" is represented as (*f0*), "Aa" for a single disease allele heterozygote by (*f1*), and "aa" for 2 disease variant homozygote by (*f2*). The additive model penetrance probability parameters were [$f0 = 0.01, f1 = 0.025, f2 = 0.04$] thus having a relative risk of 2.5 for the heterozygote and 4.0 for the homozygote double mutant. The multiplicative model parameters were [$f0 = 0.01, f1 = 0.020, f2 = 0.04$] yielding relative risk of 2.0 for the heterozygote and 4.0 for the homozygous double mutant. Both models had several levels of associated allele frequencies (0.5, 0.20, and 0.7). For individual and pooled genotyping with 0% genotype error and 0% sampling error, a sample size of 322 was needed to achieve 80% power with a significance level of $<5e^{-8}$ for the additive model. The multiplicative model needed a larger sample size, 404 to achieve the same statistical

power. Error rates of 0% (perfect genotyping and no sampling error) are a theoretical baseline for the calculations, not a practical one. When the genotyping error rate was increased to 3% (the maximum amount tested in the study), the sample size required for significance increased modestly from 322 to 376 in the additive model and from 404 to 473 samples required for the multiplicative model. These effects of genotyping error are significant, but relatively small when compared to the effects of sampling error. Introduction of sampling error in the pooled genotyping testing had a much more dramatic effect. A 1% sampling error rate increased the sample size required for the multiplicative model from 404 to 693, and the additive model form 322 to 479. Further, at a sampling error rate of 3%, Zou and Zhao claim that it is not statistically possible to achieve a $5e^{-8}$ level of significance with any sample size.

Our results somewhat agree with those of the Zou and Zhao. In general, our results show that the genotyping error rate has a modest effect on the overall significance of the genome-wide association testing, while sampling error has a dramatically larger effect. Yet, the Zou and Zhao study leaves a wide open gap in the sample size requirements needed in the presence of sampling error. Namely, the sample size required goes from 479 (additive model) and 693 (multiplicative model) to ∞ *infinity* when sampling error increases from 1% to 3%. The fact that the level of significance is set to $5e^{-8}$ implies a stringent correction for multiple testing and is arguably too stringent of a p-value cutoff [1, 64, 65]. Perhaps a less conservative p-value threshold would have served better in increasing the resolution of the Zou and Zhao study. This would have filled the immense gap occurring from the 1% to 3% sampling error rate where the conclusion is that anything more than a 1% sampling error rate is a “kiss of death” for pooled

genotyping association studies. Our study results use a less stringent cutoff of (p-value $<5e^{-2}$), although it has been argued that this threshold is not stringent enough. Given that our datasets consisted of 10,000 SNPs, a p-value of $5e^{-2}$ might yield 500 false positives by chance alone along with the true positives. For the SNP size in our study, this was determined as an acceptable amount of "noise". For larger SNP numbers in the 100,000 to 1,000,000 ranges as the current generation of genotyping platforms can achieve, a higher threshold will definitely be needed.

Our study uses a 3 locus disease model which represents more complex and more "real world" model where there are multiple genes contributing to the allelic spectra of disease, rather than additive or multiplicative effects of the heterozygous and homozygous state of a single locus. This use of single locus models for simulating complex diseases misses the broader issue in that complex diseases are due to multiple loci. This is not due to poor research design by previous groups, but rather to the lack of the proper computational tools needed to create the proper models. This was a major factor in initiating our study using a multi marker model of complex disease, and using genomeSIM as the genotype simulation package in order to achieve it.

The choice of sample size was influenced by the results of the type 2 diabetes (T2D) and AMD studies. As a higher sample sized analysis, the T2D GWA used 1,363 cases and controls to find 8 putative disease associated SNPs with low odds ratios of 1.19 to 2.77 [12]. As a lower sample sized evaluation, the AMD GWA used 146 cases and controls to yield 2 disease associated SNPs with high odds ratios of 4.6 to 7.5 [62]. These examples can be generalized to show that sample size needed for a GWA is a

function of the disease relative risk or odds ratios, the allele frequencies, and the significance level cutoff.

A schematic from a review by Wang et. al. [1] of sample size required for properly powered GWA relative to allele frequency at varying odds ratios is shown in Figure 12.

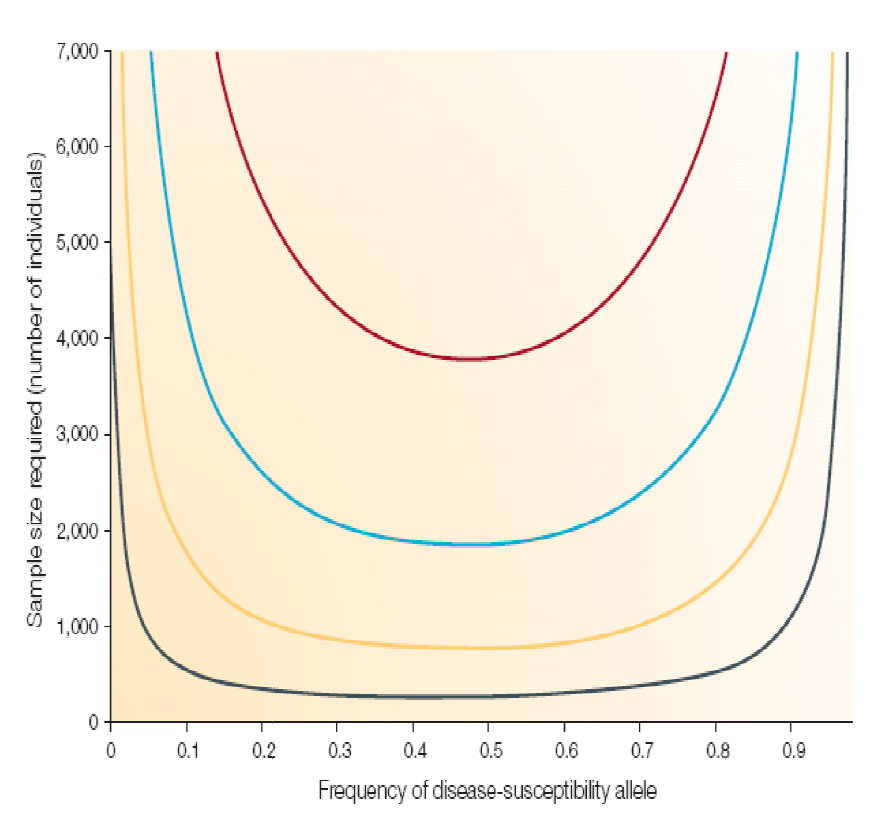


Figure 12: Sample size needed in a Genome-wide association (GWA) study.

Relative to disease allele frequency @ varying odds ratios (Wang et. al. [1]])

Statistical power of 80% at significance level of $P < 1e^{-6}$ for a multiplicative model

Odds ratios

1.2 = top curve: red

1.3 = second curve from top: blue

1.5 = third curve from top: orange

2.0 =bottom curve: green

The absolute sample size from Figure 12 is not the most important feature of the graph. Different levels of threshold significance or an additive disease model assumption instead of multiplicative would change the sample size requirements substantially. Rather, it is the shape of the curves and the degree of change from the varying odds ratios which is substantial. As the allele frequencies of the disease associated SNPs becomes smaller than 0.05 (or larger than 0.95), there is a sharp exponential increase in the required sample size. This points to the difficulty in terms of large sample sizes needed, and therefore large costs, of finding "rare" alleles by GWA. This boundary between common and rare alleles is not precisely defined, but in the context of GWA has been eluded at or near (MAF = 0.05). Indeed the HAPMAP consortium sets this figure as the cutoff between "rare" and "common" alleles [27].

The fact that resolution of rare disease associated alleles requires such large sample sizes suggests that even current large scale GWA studies having hundreds, even thousands of individuals may still not be adequately powered to detect these rare alleles. As individual genotyping is already cost prohibitive when looking for the common alleles, the potential usefulness if not necessity of pooled genotyping as a cost effective measure in GWA with large sample sizes is highlighted further.

In terms of sample size parameter selection for our simulation study, the lower sample size requirements of Figure 12 suggest that a multiplicative complex disease SNP with an odds ratio of 2.0 and MAF of 0.20 requires a sample size of 200 to 300 individuals. The lower odds ratios required much larger sample sizes, ~5,000 individuals for odds ratio 1.2 at MAF 0.20. However, the p-value cutoff in the Wang et. al. figure was conservative at p-value $<1e^{-6}$ and the disease model was multiplicative. Given the

fact that our p-value cutoff was less stringent and we were using an additive model as well as a multiplicative model, we set the low end of sample size at 200 individuals (100 cases and 100 controls). The fact that the AMD genome-wide association resolved highly penetrant markers with a 146 individual sample size reinforced the 200 sample size starting point for our simulation parameter. From there, incremental increases in sample sizes were iteratively processed until the significance threshold (p-value < $5e^{-2}$) for was surpassed at a sample size of 1,000 for pooling with a multiplicative model at the highest tested relative risk of 2.5 (Figure 6).

The sample size of 500 cases and 500 controls was used for all other sub-sections of the study because it represents a "sweet spot" or point of inflection. In terms of the relative risk ranges in our study, the sample size of 1,000 yielded association testing p-values which exceeded threshold cutoff significance at the highest relative risk range tested (2.5 - 10) and did not achieve significance at the lowest relative risk range (1.08 - 1.5). This was true for both the additive and multiplicative models and also true for both individual and pooled genotyping. Additionally, each combinatoric possibility of above or below threshold between individual and pooled genotyping is also seen at the sample size of 1,000. For example, with a sample size of 1,000 in the **multiplicative model**, pooled and individual genotyping both identified all 3 disease associated SNPs at relative risk 2.5. At relative risk 2.0, only individual genotyping resolved all 3 SNPs, while pooling found none of them. At relative risk 1.5, neither pooling nor individual genotyping were able to detect any of the SNPs above threshold. Similar results are seen in the **additive model**. All 3 SNPs were resolved by pooling and individual genotyping at relative risk 2.5 and also 2.0. At relative risk 1.5, only individual genotyping was able

to detect the 3 SNPs, while pooling found none. The SNPs fell short of the significance threshold at a relative risk of 1.33 with the additive model (Figure 7). Thus, the sample size of 1,000 was chosen because it had all of the possible combinations of above or below threshold cutoffs; for both additive and multiplicative models; and for both individual genotyping and pooling. All of this occurring within the chosen parameter ranges of relative risk 1.08 – 2.5 and MAF 0.20.

CHAPTER VI

CONCLUSIONS

The objective of this evaluation was to characterize the differences between individual genotyping and pooled allelotyping in terms of the nature of loss of resolution. This objective was further defined to include sections addressing the comparisons in the context of sampling error, genotyping error, relative risk level, sample size, and additive or multiplicative allelic disease spectrums.

It has been well supported that pooling will introduce an additional source of error into genome-wide association (GWA) analysis not present in individual genotyping; sampling error [1, 38, 39, 48]. Further, this sampling error can substantially lower the statistical power in a pooling based GWA analysis [38, 39]. The effect of sampling error in pooled genotyping for our evaluation is shown in Figure 10. Our results indicate that sampling error has a dramatic effect on the overall ability of a GWA to detect the true disease associated markers. Sampling errors of 4% and 5% in the additive complex disease model lowered the result significances to the point that none of the disease associated variants were detected above threshold. These results are not as dramatic as the Zou and Zhaou study [39] which concluded that it was not possible to achieve a significance level of (p-value<$5e^{-8}$) at 80% power in a GWA using pooled allelotyping with 3% sampling error rate. Another simulation study by Pearson et. al. showed that sampling error lowered the rank of the disease associated SNP from #1 out of 100,000 SNPs by individual genotyping to a rank >5,000 by pooling with 3% sampling error [6]. This would likely result in exclusion of the SNP out of even the most liberal p-value

($<5e^{-2}$) threshold cutoff. The same study concluded that sampling errors of 1% and 2% lowered the ranks as well, but to a much lesser degree. A 2% sampling error rate lead to the SNP still ranked within the top 1,000 and a 1% sampling error yielded a top 10 ranking. Our results are generally in agreement with the Zou and Pearson data in that sampling error is a critical factor and can be tolerated only to a small degree in pooled GWA analysis. Sampling errors of 2% to 3% will significantly impact the overall results to the point that the disease associated variants may be lost from the selection criteria altogether. A critical question arises as to overall sampling error rate and, more importantly, what minimum sampling error rates are realistically achievable in practice. A study by Kirov et. al. found that sampling errors on the Affymetrix 10k genotyping platform have a mean error rate of 1.37% [55]. Additionally, 95% of all the SNPs on the 10K have a rate of error <3.2%. This evaluation used the Affymetrix 10K Xba 142 2.0 array. This in part lead our study to used the 10,000 SNP model with a 2% sampling error rate in our study; as sampling error had been well characterized for a real 10,000 genotyping platform. The Kirov results cite that the 10K error rate is similar to that of genotyping platforms with fewer than 10,000 SNPs. Whether this error rate scales up to the 100k, 500k, and 1 million platforms by Affymetrix (as well as other commercial platforms) is unknown. There are marked differences in Affeymetrix genotyping platforms larger than 10K. Firstly, the number of quartets used to probe each SNP was reduced from 40 on the 10K array to 24 on the higher number arrays. Quartets are comprised of perfect match and a mismatch for the wild-type and variant alleles. Thus each quartet for a SNP consists of: 1) allele “A” perfect match; 2) allele “A” mismatch; 3) allele “a” perfect match; 4) allele “a” mismatch. Additionally, the feature size has

been reduced from 8 microns in the 10K array to 5 microns in the higher number SNP arrays.

Our results for the effect of genotyping error contrast to that of sampling error. We found that both individual and pooled genotyping are not as dramatically affected by genotyping error as by sampling error. Although both individual and pooled genotyping are significantly affected by genotyping errors to some degree, our results show that this is much less of a factor in overall results than the effect of sampling error on pooled allelotyping. We found that genotyping errors even up to 10% were less detrimental than a sampling error increase of only 1% (Figure 9). For the parameters tested (relative risk of 2.0 and sample size 1,000) genotyping error up to 10% did not drop any significance levels out of the threshold cutoff of $<5e^{-2}$. In contrast, a sampling error rate of 3% lessened the significance of pooled association testing to well out of the threshold range and none of the 3 disease associated variants were detectable. Future analysis may examine this with a lower level of relative risk or fewer samples as the dynamics of genotyping error may change at different levels. The evaluation did not consider genotyping error rates greater than 10% as this has been used an indication of "poor quality" DNA that has degraded. Indeed, studies exclude any results where <90% of the SNPs are called across the samples or total array platform call rates are <85% [6, 7]. The method of genotype error generation by our genotype simulator, genomeSIM, uses a uniform random distribution.

In terms of allelic spectrum of complex disease, and the differences between additive and multiplicative variant resolution for individual and pooled genotyping were

examined. Our results conclude that common diseases with an additive allele effect are more readily resolved by both individual genotyping and pooling (Figure 13).

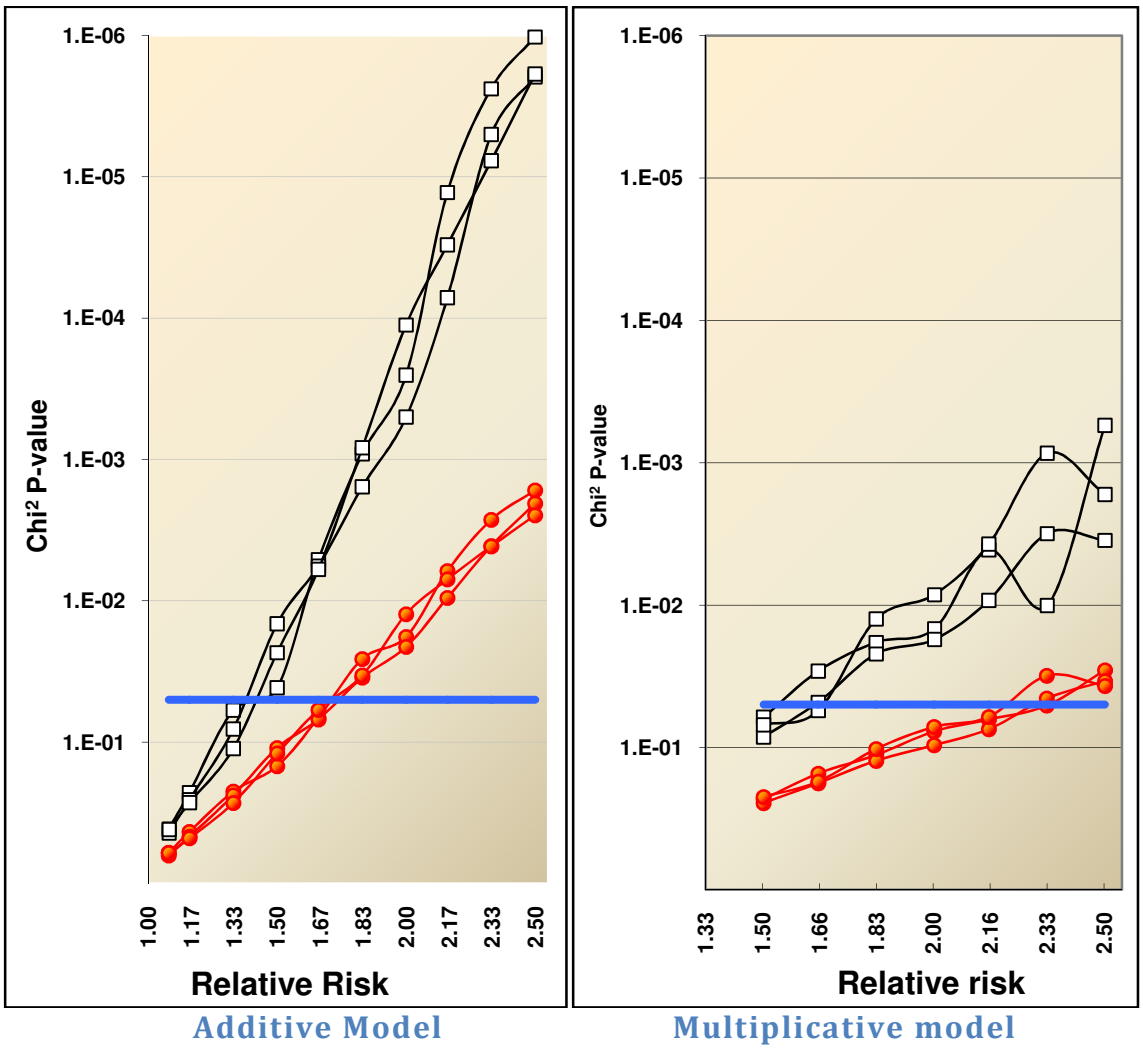


Figure 13: Individual and Pooled genotyping for Additive and Multiplicative models:

Individual genotyping: white squares with black lines:
Pooled genotyping with 2% sampling error: red circles with red line.
Cutoff threshold (p-value = 5e-2): blue horizontal line

Data points represent average p-value of 100 simulations

Both individual and pooled genotyping resolved all 3 disease associated variants at a lower minimum relative risk in the additive model than the multiplicative model. This is not surprising as other groups have suggested that additive common diseases are more amenable to GWA analysis in general [1, 16]. Looking at the graph of the allelic spectra of the disease models (Figure 2), it is clear that the non-end points of the additive model all have a higher disease probability than the corresponding multiplicative model points.

The penetrance functions for disease probability were constructed by making a 3 locus bi-allelic model. These models would have 27 possible genotype combinations, but these states would bin into 7 levels of disease penetrance corresponding to the number of disease associated alleles present (0 to 6). The phenocopy rate of 10% was the baseline in all of the penetrance functions. This 10% disease probability phenocopy represents the background non-genetic disease probability in all disease model functions where none of the disease associated variants are present in the genotype. The end-point relative risks (corresponding to 6 disease associated allele variants in the genotype) ranged from 1.5 to 10.0. They were set to cover the ranges of relative risk or odds ratios seen in recent GWA analysis[2]. For a low end odds ratios, type 2 diabetes alleles having odds ratios of 1.19 – 1.3 are cited as examples [12-14, 47]. As a high end of odds ratios, the AMD data with odds ratios of 5 to 7 are cited as examples [62, 66, 67] as well as Alzheimer's ApoE-ε4 with odds ratios of 3 to 8 [6].

Our penetrance functions were created with relative risks as opposed to odds ratios. This was done in order to make the results more general and therefore applicable to a range of complex disease characteristics. The **relative risk** is defined as the ratio of

disease probability given the disease genotype over the probability of disease given the disease genotype is not present. For example a wild-type genotype with disease probability of 0.10 and a disease genotype having 1 disease associated SNP and a disease probability of 0.20 will have a relative risk of 2.0. The **odds ratio** is defined as the odds of disease given disease genotype over the odds of disease given no disease genotype. Using the same probabilities as the previous example, the odds of disease an individual with a disease genotype is calculated as the probability of having the disease given disease genotype over the probability of not having disease given the disease genotype, $[0.20/(1-0.20)) = 0.20/0.80]$. The odds of disease in an individual with wild-type genotype are $[0.10/0.90]$. And the odds ratio is the ratio of odds of disease in an individual with the disease genotype over the odds of disease in an individual with the wild-type genotype $[(20/80)/(10/90)] = 2.25$. Odds ratio and relative risk asymptotically approach each other as they become close to 1, and as such were used somewhat interchangeably relating GWA results from the field with our study at the levels of relative risk tested. Odds ratios are more informative than relative risk in specific disease examples because they take the odds of disease and non-disease into account. However, relative risk is a more general measure and more appropriate for our study as the objective is to examine the general case of common disease rather than a specific disease with known odds in the population.

The result was the creation of 20 penetrance function to represent additive and multiplicative models which low range from relative risks of 1.07 for a single disease SNPs to 1.5 for 6 disease SNPs; to a high range of relative risk 2.5 for a single disease variant to 10.0 for 6 SNPs. The multiplicative functions, by definition, will have lower

penetrance probabilities in the genotypes with 1 to 5 disease associated SNPs because they have the same 0 and 6 disease allele penetrance probabilities as the additive functions. The result is that the causal variants created using the multiplicative model penetrance functions were less resolvable than the additive model template. Further, the gap in minimum resolvable relative risk between individual genotyping and pooled genotyping was twice as large in the multiplicative model than in the additive model. These results suggest that pooled allelotyping is much less viable in identifying common disease associated loci with a multiplicative effect characteristic.

The choice of a liberal p-value cutoff threshold of $5e^{-2}$ was made in the context of passing on an acceptable number of putative candidate associations for further analysis. This p-value threshold will falsely identify ~500 SNPs from 10,000 SNPs tested as disease associated when they are actually due to chance alone. Other groups have argued for the use of a Bonferroni correction to account for multiple testing [22]. Indeed, a p-value of $<5e^{-2}$ is useless for genotyping platforms with one million or even 100,000 SNPs. For example, using this p-value on a 1,000,000 SNP platform would result in ~50,000 false positives. However, Bonferroni assumes independence of SNPs. Given the fact that 70-80% of the human genome is in strong LD [27], the assumption of SNP independence does not hold and as a result Bonferroni becomes over stringent. Additionally, if Bonferroni is to be used, the sample size required to achieve this "lofty" significance level becomes very large. Alternative strategies have been proposed to access proper significance levels such as a Bayesian approach toward determining the likelihood a true association [68]. Alternatively, permutation testing offers a good solution to empirically assessing the probability of having observed a particular result by

chance[16]. Using the association test statistic, a threshold for significance in whole-genome association analysis can be derived empirically. Thus, permutation testing can be conducted where the status of case or control is randomized in the population and a p-value is derived from the association test statistic values [69]. The question of proper significance levels for GWA studies remains the subject of current research and debate.

It has been suggested that a multiphase hybrid design could incorporate pooling as the first round "broad" sweep in a genome wide association scan and subsequent individual genotyping on candidate regions or SNPs [1, 16, 38]. In this way, the cost effective gain in efficiency could be realized by using pooling to generate a list of putative associated SNPs. Even if many of the "associated" SNPs are false positives, later rounds of testing would benefit with fewer SNPs to test via the pooled information. Our study further supports the position that pooled allelotyping is not a viable replacement for individual genotyping, but has the real potential as a useful screening tool at the whole-genome level.

Our work has created a system which allows for the direct evaluation and comparison of pooled allelotyping versus individual genotyping for genome-wide association analysis of complex disease. This was done by using existing bioinformatics tools; GenomeSIM for genotype simulations of complex disease [49]; Haploview for individual genotyping [50]; and an extensively modified version of the Pooled DNA analyzer [51] termed sm_PDA for pooled allelotyping. Additionally, MATLAB scripts to process and parse the files allowed the tools to be connected as a single process. It is clear that pooled allelotyping for genome-wide association studies has both benefits and disadvantages. To be successful, a study will have to carefully weigh these

considerations in the experimental design phase. The potential savings of time and money due to less overall genotyping must be carefully weighed against the degree and characteristics of power and information reduced in the pooling approach. Our system allows for these considerations to be assessed, evaluated, and compared.

APPENDIX A: PENETRANCE TABLES AND MATLAB CODE

Penetrance Table 1 for Relative Risk Range 1.08 - 1.50

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	**B**	**C**	**Genotype**	**Additive**	**Multiplicative**
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.1083	0.1070
AA	Bb	CC	AABbCC	0.1083	0.1070
AA	BB	Cc	AABBCc	0.1083	0.1070
aa	BB	CC	aaBBCC	0.1167	0.1145
Aa	Bb	CC	AaBbCC	0.1167	0.1145
AA	bb	CC	AAbbCC	0.1167	0.1145
Aa	BB	Cc	AaBBCc	0.1167	0.1145
AA	Bb	Cc	AABbCc	0.1167	0.1145
AA	BB	cc	AABBcc	0.1167	0.1145
aa	Bb	CC	aaBbCC	0.1250	0.1225
Aa	bb	CC	AabbCC	0.1250	0.1225
aa	BB	Cc	aaBBCc	0.1250	0.1225
Aa	Bb	Cc	AaBbCc	0.1250	0.1225
AA	bb	Cc	AAbbCc	0.1250	0.1225
Aa	BB	cc	AaBBcc	0.1250	0.1225
AA	Bb	cc	AABbcc	0.1250	0.1225
aa	bb	CC	aabbCC	0.1333	0.1310
aa	Bb	Cc	aaBbCc	0.1333	0.1310
Aa	bb	Cc	AabbCc	0.1333	0.1310
aa	BB	cc	aaBBcc	0.1333	0.1310
Aa	Bb	cc	AaBbcc	0.1333	0.1310
AA	bb	cc	AAbbcc	0.1333	0.1310
aa	bb	Cc	aabbCc	0.1417	0.1402
aa	Bb	cc	aaBbcc	0.1417	0.1402
Aa	bb	cc	Aabbcc	0.1417	0.1402
aa	bb	cc	aabbcc	0.1500	0.1500

Penetrance Table 2 for Relative Risk 1.16 – 2.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				*Disease Risk (Probability)*	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.1167	0.1122
AA	Bb	CC	AABbCC	0.1167	0.1122
AA	BB	Cc	AABBCc	0.1167	0.1122
aa	BB	CC	aaBBCC	0.1333	0.1260
Aa	Bb	CC	AaBbCC	0.1333	0.1260
AA	bb	CC	AAbbCC	0.1333	0.1260
Aa	BB	Cc	AaBBCc	0.1333	0.1260
AA	Bb	Cc	AABbCc	0.1333	0.1260
AA	BB	cc	AABBcc	0.1333	0.1260
aa	Bb	CC	aaBbCC	0.1500	0.1414
Aa	bb	CC	AabbCC	0.1500	0.1414
aa	BB	Cc	aaBBCc	0.1500	0.1414
Aa	Bb	Cc	AaBbCc	0.1500	0.1414
AA	bb	Cc	AAbbCc	0.1500	0.1414
Aa	BB	cc	AaBBcc	0.1500	0.1414
AA	Bb	cc	AABbcc	0.1500	0.1414
aa	bb	CC	aabbCC	0.1667	0.1587
aa	Bb	Cc	aaBbCc	0.1667	0.1587
Aa	bb	Cc	AabbCc	0.1667	0.1587
aa	BB	cc	aaBBcc	0.1667	0.1587
Aa	Bb	cc	AaBbcc	0.1667	0.1587
AA	bb	cc	AAbbcc	0.1667	0.1587
aa	bb	Cc	aabbCc	0.1833	0.1782
aa	Bb	cc	aaBbcc	0.1833	0.1782
Aa	bb	cc	Aabbcc	0.1833	0.1782
aa	bb	cc	aabbcc	0.2000	0.2000

Penetrance Table 3 for Relative Risk 1.33 – 3.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.1333	0.1201
AA	Bb	CC	AABbCC	0.1333	0.1201
AA	BB	Cc	AABBCc	0.1333	0.1201
aa	BB	CC	aaBBCC	0.1667	0.1442
Aa	Bb	CC	AaBbCC	0.1667	0.1442
AA	bb	CC	AAbbCC	0.1667	0.1442
Aa	BB	Cc	AaBBCc	0.1667	0.1442
AA	Bb	Cc	AABbCc	0.1667	0.1442
AA	BB	cc	AABBcc	0.1667	0.1442
aa	Bb	CC	aaBbCC	0.2000	0.1732
Aa	bb	CC	AabbCC	0.2000	0.1732
aa	BB	Cc	aaBBCc	0.2000	0.1732
Aa	Bb	Cc	AaBbCc	0.2000	0.1732
AA	bb	Cc	AAbbCc	0.2000	0.1732
Aa	BB	cc	AaBBcc	0.2000	0.1732
AA	Bb	cc	AABbcc	0.2000	0.1732
aa	bb	CC	aabbCC	0.2333	0.2080
aa	Bb	Cc	aaBbCc	0.2333	0.2080
Aa	bb	Cc	AabbCc	0.2333	0.2080
aa	BB	cc	aaBBcc	0.2333	0.2080
Aa	Bb	cc	AaBbcc	0.2333	0.2080
AA	bb	cc	AAbbcc	0.2333	0.2080
aa	bb	Cc	aabbCc	0.2667	0.2498
aa	Bb	cc	aaBbcc	0.2667	0.2498
Aa	bb	cc	Aabbcc	0.2667	0.2498
aa	bb	cc	aabbcc	0.3000	0.3000

Penetrance Table 4 for Relative Risk 1.5 – 4.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.1500	0.1260
AA	Bb	CC	AABbCC	0.1500	0.1260
AA	BB	Cc	AABBCc	0.1500	0.1260
aa	BB	CC	aaBBCC	0.2000	0.1587
Aa	Bb	CC	AaBbCC	0.2000	0.1587
AA	bb	CC	AAbbCC	0.2000	0.1587
Aa	BB	Cc	AaBBCc	0.2000	0.1587
AA	Bb	Cc	AABbCc	0.2000	0.1587
AA	BB	cc	AABBcc	0.2000	0.1587
aa	Bb	CC	aaBbCC	0.2500	0.2000
Aa	bb	CC	AabbCC	0.2500	0.2000
aa	BB	Cc	aaBBCc	0.2500	0.2000
Aa	Bb	Cc	AaBbCc	0.2500	0.2000
AA	bb	Cc	AAbbCc	0.2500	0.2000
Aa	BB	cc	AaBBcc	0.2500	0.2000
AA	Bb	cc	AABbcc	0.2500	0.2000
aa	bb	CC	aabbCC	0.3000	0.2520
aa	Bb	Cc	aaBbCc	0.3000	0.2520
Aa	bb	Cc	AabbCc	0.3000	0.2520
aa	BB	cc	aaBBcc	0.3000	0.2520
Aa	Bb	cc	AaBbcc	0.3000	0.2520
AA	bb	cc	AAbbcc	0.3000	0.2520
aa	bb	Cc	aabbCc	0.3500	0.3175
aa	Bb	cc	aaBbcc	0.3500	0.3175
Aa	bb	cc	Aabbcc	0.3500	0.3175
aa	bb	cc	aabbcc	0.4000	0.4000

Penetrance Table 5 for Relative Risk 1.66 – 5.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.1667	0.1308
AA	Bb	CC	AABbCC	0.1667	0.1308
AA	BB	Cc	AABBCc	0.1667	0.1308
aa	BB	CC	aaBBCC	0.2333	0.1710
Aa	Bb	CC	AaBbCC	0.2333	0.1710
AA	bb	CC	AAbbCC	0.2333	0.1710
Aa	BB	Cc	AaBBCc	0.2333	0.1710
AA	Bb	Cc	AABbCc	0.2333	0.1710
AA	BB	cc	AABBcc	0.2333	0.1710
aa	Bb	CC	aaBbCC	0.3000	0.2236
Aa	bb	CC	AabbCC	0.3000	0.2236
aa	BB	Cc	aaBBCc	0.3000	0.2236
Aa	Bb	Cc	AaBbCc	0.3000	0.2236
AA	bb	Cc	AAbbCc	0.3000	0.2236
Aa	BB	cc	AaBBcc	0.3000	0.2236
AA	Bb	cc	AABbcc	0.3000	0.2236
aa	bb	CC	aabbCC	0.3667	0.2924
aa	Bb	Cc	aaBbCc	0.3667	0.2924
Aa	bb	Cc	AabbCc	0.3667	0.2924
aa	BB	cc	aaBBcc	0.3667	0.2924
Aa	Bb	cc	AaBbcc	0.3667	0.2924
AA	bb	cc	AAbbcc	0.3667	0.2924
aa	bb	Cc	aabbCc	0.4333	0.3824
aa	Bb	cc	aaBbcc	0.4333	0.3824
Aa	bb	cc	Aabbcc	0.4333	0.3824
aa	bb	cc	aabbcc	0.5000	0.5000

Penetrance Table 6 for Relative Risk 1.83 – 6.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.1833	0.1348
AA	Bb	CC	AABbCC	0.1833	0.1348
AA	BB	Cc	AABBCc	0.1833	0.1348
aa	BB	CC	aaBBCC	0.2667	0.1817
Aa	Bb	CC	AaBbCC	0.2667	0.1817
AA	bb	CC	AAbbCC	0.2667	0.1817
Aa	BB	Cc	AaBBCc	0.2667	0.1817
AA	Bb	Cc	AABbCc	0.2667	0.1817
AA	BB	cc	AABBcc	0.2667	0.1817
aa	Bb	CC	aaBbCC	0.3500	0.2449
Aa	bb	CC	AabbCC	0.3500	0.2449
aa	BB	Cc	aaBBCc	0.3500	0.2449
Aa	Bb	Cc	AaBbCc	0.3500	0.2449
AA	bb	Cc	AAbbCc	0.3500	0.2449
Aa	BB	cc	AaBBcc	0.3500	0.2449
AA	Bb	cc	AABbcc	0.3500	0.2449
aa	bb	CC	aabbCC	0.4333	0.3302
aa	Bb	Cc	aaBbCc	0.4333	0.3302
Aa	bb	Cc	AabbCc	0.4333	0.3302
aa	BB	cc	aaBBcc	0.4333	0.3302
Aa	Bb	cc	AaBbcc	0.4333	0.3302
AA	bb	cc	AAbbcc	0.4333	0.3302
aa	bb	Cc	aabbCc	0.5167	0.4451
aa	Bb	cc	aaBbcc	0.5167	0.4451
Aa	bb	cc	Aabbcc	0.5167	0.4451
aa	bb	cc	aabbcc	0.6000	0.6000

Penetrance Table 7 for Relative Risk 2.0 – 7.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.2000	0.1383
AA	Bb	CC	AABbCC	0.2000	0.1383
AA	BB	Cc	AABBCc	0.2000	0.1383
aa	BB	CC	aaBBCC	0.3000	0.1913
Aa	Bb	CC	AaBbCC	0.3000	0.1913
AA	bb	CC	AAbbCC	0.3000	0.1913
Aa	BB	Cc	AaBBCc	0.3000	0.1913
AA	Bb	Cc	AABbCc	0.3000	0.1913
AA	BB	cc	AABBcc	0.3000	0.1913
aa	Bb	CC	aaBbCC	0.4000	0.2646
Aa	bb	CC	AabbCC	0.4000	0.2646
aa	BB	Cc	aaBBCc	0.4000	0.2646
Aa	Bb	Cc	AaBbCc	0.4000	0.2646
AA	bb	Cc	AAbbCc	0.4000	0.2646
Aa	BB	cc	AaBBcc	0.4000	0.2646
AA	Bb	cc	AABbcc	0.4000	0.2646
aa	bb	CC	aabbCC	0.5000	0.3659
aa	Bb	Cc	aaBbCc	0.5000	0.3659
Aa	bb	Cc	AabbCc	0.5000	0.3659
aa	BB	cc	aaBBcc	0.5000	0.3659
Aa	Bb	cc	AaBbcc	0.5000	0.3659
AA	bb	cc	AAbbcc	0.5000	0.3659
aa	bb	Cc	aabbCc	0.6000	0.5061
aa	Bb	cc	aaBbcc	0.6000	0.5061
Aa	bb	cc	Aabbcc	0.6000	0.5061
aa	bb	cc	aabbcc	0.7000	0.7000

Penetrance Table 8 for Relative Risk 2.16 – 8.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.2167	0.1414
AA	Bb	CC	AABbCC	0.2167	0.1414
AA	BB	Cc	AABBCc	0.2167	0.1414
aa	BB	CC	aaBBCC	0.3333	0.2000
Aa	Bb	CC	AaBbCC	0.3333	0.2000
AA	bb	CC	AAbbCC	0.3333	0.2000
Aa	BB	Cc	AaBBCc	0.3333	0.2000
AA	Bb	Cc	AABbCc	0.3333	0.2000
AA	BB	cc	AABBcc	0.3333	0.2000
aa	Bb	CC	aaBbCC	0.4500	0.2828
Aa	bb	CC	AabbCC	0.4500	0.2828
aa	BB	Cc	aaBBCc	0.4500	0.2828
Aa	Bb	Cc	AaBbCc	0.4500	0.2828
AA	bb	Cc	AAbbCc	0.4500	0.2828
Aa	BB	cc	AaBBcc	0.4500	0.2828
AA	Bb	cc	AABbcc	0.4500	0.2828
aa	bb	CC	aabbCC	0.5667	0.4000
aa	Bb	Cc	aaBbCc	0.5667	0.4000
Aa	bb	Cc	AabbCc	0.5667	0.4000
aa	BB	cc	aaBBcc	0.5667	0.4000
Aa	Bb	cc	AaBbcc	0.5667	0.4000
AA	bb	cc	AAbbcc	0.5667	0.4000
aa	bb	Cc	aabbCc	0.6833	0.5657
aa	Bb	cc	aaBbcc	0.6833	0.5657
Aa	bb	cc	Aabbcc	0.6833	0.5657
aa	bb	cc	aabbcc	0.8000	0.8000

Penetrance Table 9 for Relative Risk 2.33 – 9.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.2333	0.1442
AA	Bb	CC	AABbCC	0.2333	0.1442
AA	BB	Cc	AABBCc	0.2333	0.1442
aa	BB	CC	aaBBCC	0.3667	0.2080
Aa	Bb	CC	AaBbCC	0.3667	0.2080
AA	bb	CC	AAbbCC	0.3667	0.2080
Aa	BB	Cc	AaBBCc	0.3667	0.2080
AA	Bb	Cc	AABbCc	0.3667	0.2080
AA	BB	cc	AABBcc	0.3667	0.2080
aa	Bb	CC	aaBbCC	0.5000	0.3000
Aa	bb	CC	AabbCC	0.5000	0.3000
aa	BB	Cc	aaBBCc	0.5000	0.3000
Aa	Bb	Cc	AaBbCc	0.5000	0.3000
AA	bb	Cc	AAbbCc	0.5000	0.3000
Aa	BB	cc	AaBBcc	0.5000	0.3000
AA	Bb	cc	AABbcc	0.5000	0.3000
aa	bb	CC	aabbCC	0.6333	0.4327
aa	Bb	Cc	aaBbCc	0.6333	0.4327
Aa	bb	Cc	AabbCc	0.6333	0.4327
aa	BB	cc	aaBBcc	0.6333	0.4327
Aa	Bb	cc	AaBbcc	0.6333	0.4327
AA	bb	cc	AAbbcc	0.6333	0.4327
aa	bb	Cc	aabbCc	0.7667	0.6240
aa	Bb	cc	aaBbcc	0.7667	0.6240
Aa	bb	cc	Aabbcc	0.7667	0.6240
aa	bb	cc	aabbcc	0.9000	0.9000

Penetrance Table 10 for Relative Risk 2.5 – 10.0

Uppercase [A, B, or C] denotes non-disease associated allele
Lowercase [a, b, or c] denotes disease associated variant allele

Heterozygotes for single disease variant locus highlighted in yellow
Homozygotes for 2 disease variant allele locus highlighted in blue

Disease Associated Locus				Disease Risk (Probability)	
A	***B***	***C***	***Genotype***	***Additive***	***Multiplicative***
AA	BB	CC	AABBCC	0.1000	0.1000
Aa	BB	CC	AaBBCC	0.2500	0.1468
AA	Bb	CC	AABbCC	0.2500	0.1468
AA	BB	Cc	AABBCc	0.2500	0.1468
aa	BB	CC	aaBBCC	0.4000	0.2154
Aa	Bb	CC	AaBbCC	0.4000	0.2154
AA	bb	CC	AAbbCC	0.4000	0.2154
Aa	BB	Cc	AaBBCc	0.4000	0.2154
AA	Bb	Cc	AABbCc	0.4000	0.2154
AA	BB	cc	AABBcc	0.4000	0.2154
aa	Bb	CC	aaBbCC	0.5500	0.3162
Aa	bb	CC	AabbCC	0.5500	0.3162
aa	BB	Cc	aaBBCc	0.5500	0.3162
Aa	Bb	Cc	AaBbCc	0.5500	0.3162
AA	bb	Cc	AAbbCc	0.5500	0.3162
Aa	BB	cc	AaBBcc	0.5500	0.3162
AA	Bb	cc	AABbcc	0.5500	0.3162
aa	bb	CC	aabbCC	0.7000	0.4642
aa	Bb	Cc	aaBbCc	0.7000	0.4642
Aa	bb	Cc	AabbCc	0.7000	0.4642
aa	BB	cc	aaBBcc	0.7000	0.4642
Aa	Bb	cc	AaBbcc	0.7000	0.4642
AA	bb	cc	AAbbcc	0.7000	0.4642
aa	bb	Cc	aabbCc	0.8500	0.6813
aa	Bb	cc	aaBbcc	0.8500	0.6813
Aa	bb	cc	Aabbcc	0.8500	0.6813
aa	bb	cc	aabbcc	1.0000	1.0000

CODE: GENOMESIM_2_PDA.M

```
function [PoolAF SnpName IndPI]  = genomeSIM2PDA(in)

%input (in) = 10,001(1 disease state + 10k SNP)
                        by 200 - 1000 (100-500cases / 100-500 controls) matrix:
%column 1 = Major Allele Freq for Controls
%column 2 = Minor Allele Freq for Controls
%column 3 = Major AF for Cases
%column 4 - Minor AF for Cases

%output (out) = (number_of_SNPS+1) by 5 matrix
%   column1 = disease state (1 for control || 2 for case)
%   column2 = SNP number (name)
%   column3 = number of samples in pool (cases || controls)
%   column4 = Major AF
%   column5 = Minor AF
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

[number_of_samples number_of_SNPS] = size(in);
global number_of_cases;
number_of_cases = length(find(in(:,1)==1));
global number_of_controls;
number_of_controls = length(find(in(:,1)==0));
alellefreq = zeros(4,number_of_SNPS);
controlfreq = zeros(2,number_of_SNPS);
casefreq = zeros(2,number_of_SNPS);
for sample_row = 1:number_of_samples
   if in(sample_row,1)==0 %control sample (non-disease)
      for ii = 1:number_of_SNPS
         if in(sample_row,ii)==0 %'aa' = homozygous little 'a'
            controlfreq(1,ii)=controlfreq(1,ii)+2;
         elseif in(sample_row,ii)==1 %'Aa' = heterozygous
            controlfreq(1,ii)=controlfreq(1,ii)+1;
            controlfreq(2,ii)=controlfreq(2,ii)+1;
         elseif in(sample_row,ii)==2%'AA' = homozygous big 'A'
            controlfreq(2,ii)=controlfreq(2,ii)+2;
         end
      end
   elseif in(sample_row,1)==1 %case sample (disease)
      for ii = 1:number_of_SNPS
         if in(sample_row,ii)==0
            casefreq(1,ii)=casefreq(1,ii)+2;
         elseif in(sample_row,ii)==1
            casefreq(1,ii)=casefreq(1,ii)+1;
            casefreq(2,ii)=casefreq(2,ii)+1;
         elseif in(sample_row,ii)==2
            casefreq(2,ii)=casefreq(2,ii)+2;
         end
      end
   end
end
for jj = 1:number_of_SNPS
   alellefreq(1,jj) = controlfreq(1,jj)/(controlfreq(1,jj)+ controlfreq(2,jj));
   alellefreq(2,jj) = controlfreq(2,jj)/(controlfreq(1,jj)+ controlfreq(2,jj));
```

```
    alellefreq(3,jj) = casefreq(1,jj)/(casefreq(1,jj)+ casefreq(2,jj));
    alellefreq(4,jj) = casefreq(2,jj)/(casefreq(1,jj)+ casefreq(2,jj));
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

alellefreq = alellefreq';
out_temp = zeros(number_of_SNPS*2,5);
PoolAF = zeros(number_of_SNPS*2,5);
SnpName = zeros(number_of_SNPS,1);
IndPI = ones(number_of_SNPS,4);

% Create column1 = 1 for control, 2 for case
out_temp(1:number_of_SNPS,1) = 1;
out_temp(number_of_SNPS+1:end,1) = 2;

% Create column2 = SNP number
for ii = 1:number_of_SNPS
    out_temp(ii,2) = ii;
end
for jj = 1:number_of_SNPS
    out_temp(number_of_SNPS+jj,2) = jj;
end

% Create column3 = number of samples in pool
out_temp(1:number_of_SNPS,3) = number_of_cases;
out_temp(number_of_SNPS+1:end,3) = number_of_controls;

% create cols 4 - allele freq controls
out_temp(1:number_of_SNPS,4) = alellefreq(:,1);
out_temp(number_of_SNPS+1:end,4) = alellefreq(:,3);
out_temp(1:number_of_SNPS,5) = alellefreq(:,2);
out_temp(number_of_SNPS+1:end,5) = alellefreq(:,4);

PoolAF = out_temp;
SnpName = out_temp(1:number_of_SNPS,2);
IndPI(:,2) = SnpName(1:number_of_SNPS,1);
```

CODE: GENOMESIM_2_HAPLOVIEW.M

```
function [haploviewPED haploviewINF] = genomeSIM2haploview(in)

[samples SNPs] = size(in);
out=zeros(samples,SNPs*2);
haploview_header = zeros(samples,6);
haploviewINF = zeros(samples,2);
kk = 1; %column counter for 0 and 1 of genotypes
for ii = 1:samples
   if mod(ii,100)==0
      fprintf('\ndone with %d samples\n',ii)
   end
   for jj = 1:SNPs % row counter
      if in(ii,jj)==0
         out(ii,kk) = 1;
         out(ii,kk+1)=1;
      elseif in(ii,jj)==1
         out(ii,kk)=1;
         out(ii,kk+1)=2;
      elseif in(ii,jj)==2
         out(ii,kk)=2;
         out(ii,kk+1)=2;
      end %for jj
      kk = kk+2; % go forward 2 cols
   end % for ii
   kk = 1; %reset out array to begining col of next column
end % for kk

%create first 6 columns for HAPLOVIEW header
for qq = 1:samples
   haploview_header(qq,1) = qq; %sample number
   haploview_header(qq,2) = 525; %alphanumeric ID for family name
   %columns 3 and 4 are for father and mother ID in family studies
   haploview_header(qq,5) = 1; %1 = male , 2 = female
   haploview_header(qq,6) = in(qq,1)+1; % gSIM infile has (0 = control)
   % and (1 = case) haploview format needs (1 = case), (2 = control)
end
for ww = 1:SNPs
   haploviewINF(ww,1:2)=ww;
end
clear in;
format short;
fprintf('\ntime_to_calculate%d \n',toc);
haploviewPED = cat(2,haploview_header,out);
fprintf('\ndone_genomeSIM2haploview');
% dlmwrite('haploview.ped',haploviewPED,'delimiter','\t');
% dlmwrite('haploview.inf',haploviewINF,'delimiter','\t');
% fprintf('time_to_DLM write%d \n',toc);
```

CODE: PDA_2_PVAL.M:

```
function automate_PDA2pval_maf3(error)

rank_matrix = [];
pval_all_rr_vals=[];
stdev_pval_all_rr_vals=[];

for RR=[4 7 10]
   for ind = 500
      for rep = 1:100
         load_it = sprintf('load PDA_cpr_add_RR%d_maf3_rep%d_se%d.txt',RR,rep,error);
         eval(load_it);
         rank_fname = sprintf('PDA_cpr_add_RR%d_maf3_rep%d_se%d',RR,rep,error);
         a = eval(rank_fname);
         rank_matrix = cat(1,rank_matrix,a);
      end %for rep

      % sort 100 rep matrix of RR by pval (col#3)
      sorted_by_pval = sortrows(rank_matrix,[1 3]);
      %sorted_by_pval_fname=sprintf('PDA_cpr_matrix_multi_RR%d_ind%d_se%d.txt',RR,ind,error);
      %dlmwrite(sorted_by_pval_fname,sorted_by_pval,'delimiter','\t');

      %pull out causal SNPs and average their pval (col #3) of
      SNP_6_sum = 0;SNP_11_sum = 0;SNP_16_sum = 0;
      SNP_6_count = 0;SNP_11_count = 0;SNP_16_count = 0;
      for ii = 1:size(sorted_by_pval,1)
         if sorted_by_pval(ii,1)==6
            SNP_6_sum = SNP_6_sum + sorted_by_pval(ii,3);
            SNP_6_count = SNP_6_count+1;
         elseif sorted_by_pval(ii,1)==11
            SNP_11_sum = SNP_11_sum + sorted_by_pval(ii,3);
            SNP_11_count = SNP_11_count+1;
         elseif sorted_by_pval(ii,1)==16
            SNP_16_sum = SNP_16_sum + sorted_by_pval(ii,3);
            SNP_16_count = SNP_16_count+1;
         end %sorted_by_pval
      end % for ii

      % pvals for this RR set of 100 reps
      stdev_SNPa = std(sorted_by_pval(1:100,3));
      stdev_SNPb = std(sorted_by_pval(101:200,3));
      stdev_SNPc = std(sorted_by_pval(201:300,3));
      average_pval_temp =
[SNP_6_sum/SNP_6_count;SNP_11_sum/SNP_11_count;SNP_16_sum/SNP_16_count];
      average_pval_fname = sprintf('PDA_average_pval_add_RR%d_maf3_se%d.txt',RR,error);
      stdev_pval_temp = [stdev_SNPa; stdev_SNPb; stdev_SNPc];
      stdev_pval_fname = sprintf('PDA_stdev_pval_add_RR%d_maf3_se%d.txt',RR,error);
      dlmwrite(average_pval_fname, average_pval_temp);
      dlmwrite(stdev_pval_fname, stdev_pval_temp);

      % concat data with all RR samples
      pval_all_rr_vals = cat(2,pval_all_rr_vals,average_pval_temp);
      stdev_pval_all_rr_vals = cat(2,stdev_pval_all_rr_vals,stdev_pval_temp);
      rank_matrix = [];
```

```
    end % for ind

    final_pval_fname = sprintf('final_pval_add_RR%d_maf3_se%d.txt',RR,error);
    dlmwrite(final_pval_fname,pval_all_rr_vals);
    final_stdev_pval_fname = sprintf('final_pval_stdev_add_RR%d_maf3_se%d.txt',RR,error);
    %eval(final_pval_fname);
    dlmwrite(final_stdev_pval_fname,stdev_pval_all_rr_vals);
end %for RR
```

CODE: HAPLOVIEW_2_PVAL.M

```
function automate_HAP2pval.m

rank_matrix = [];
rank_all_rr_vals=[];
pval_all_rr_vals=[];
stdev_pval_all_rr_vals=[];

for param_1 = 0:5
   %load 100 repition values of RR set into matrix
   for rep = 1:100
      load_it = sprintf('load Hap_pval_rank_ge%d_%d.txt',param_1,rep);
      eval(load_it);
      rank_fname = sprintf('Hap_pval_rank_ge%d_%d',param_1,rep);
      a = eval(rank_fname);
      rank_matrix = cat(1,rank_matrix,a);
   end %for rep

%     % sort 100 rep matrix of RR by rank
%     sorted_by_rank = sortrows(rank_matrix,4);
%     sorted_by_rank_fname=sprintf('HAP_sorted_rank_RR%d.txt',param_1);
%     dlmwrite(sorted_by_rank_fname,sorted_by_rank,'delimiter','\t');

   %pull out causal SNPs and average their RANKS (col #3) of
   %rank matrix
   SNP_6_sum = 0;SNP_11_sum = 0;SNP_16_sum = 0;
   SNP_6_count = 0;SNP_11_count = 0;SNP_16_count = 0;
   for ii = 1:size(rank_matrix,1)
      if rank_matrix(ii,1)==6
         SNP_6_sum = SNP_6_sum + rank_matrix(ii,3);
         SNP_6_count = SNP_6_count+1;
      elseif rank_matrix(ii,1)==11
         SNP_11_sum = SNP_11_sum + rank_matrix(ii,3);
         SNP_11_count = SNP_11_count+1;
      elseif rank_matrix(ii,1)==16
         SNP_16_sum = SNP_16_sum + rank_matrix(ii,3);
         SNP_16_count = SNP_16_count+1;
      end %if rank_matrix
   end %for ii
%
   average_ranks_temp =
[SNP_6_sum/SNP_6_count;SNP_11_sum/SNP_11_count;SNP_16_sum/SNP_16_count];
   average_ranks_fname = sprintf('HAP_average_rank_ge%d.txt',param_1);
   dlmwrite(average_ranks_fname, average_ranks_temp);
   rank_all_rr_vals = cat(2,rank_all_rr_vals,average_ranks_temp);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%     % sort 100 rep matrix of RR by pval (col#3)
%     sorted_by_pval = sortrows(rank_matrix,3);
%     sorted_by_pval_fname = sprintf('HAP_sorted_pval_RR%d.txt',param_1);
%     dlmwrite(sorted_by_pval_fname,sorted_by_pval,'delimiter','\t');

   %pull out causal SNPs and average their pval (col #3) of
   %'sorted_by_pval' matrix
```

```
    SNP_6_sum = 0;SNP_11_sum = 0;SNP_16_sum = 0;
    SNP_6_count = 0;SNP_11_count = 0;SNP_16_count = 0;
    for ii = 1:size(rank_matrix,1)
        if rank_matrix(ii,1)==6
            SNP_6_sum = SNP_6_sum + rank_matrix(ii,2);
            SNP_6_count = SNP_6_count+1;
        elseif rank_matrix(ii,1)==11
            SNP_11_sum = SNP_11_sum + rank_matrix(ii,2);
            SNP_11_count = SNP_11_count+1;
        elseif rank_matrix(ii,1)==16
            SNP_16_sum = SNP_16_sum + rank_matrix(ii,2);
            SNP_16_count = SNP_16_count+1;
        end %rank_matrix
    end % for ii
    % pvals for this RR set of 100 reps
    stdev_SNPa = std(rank_matrix(1:100,3));
    stdev_SNPb = std(rank_matrix(101:200,3));
    stdev_SNPc = std(rank_matrix(201:300,3));
    average_pval_temp =
[SNP_6_sum/SNP_6_count;SNP_11_sum/SNP_11_count;SNP_16_sum/SNP_16_count];
    average_pval_fname = sprintf('HAP_pval_ge%d.txt',param_1);
    stdev_pval_temp = [stdev_SNPa; stdev_SNPb; stdev_SNPc];
    stdev_pval_fname = sprintf('HAP_pval_stdev_ge%d.txt',param_1);
    dlmwrite(average_pval_fname, average_pval_temp);
    dlmwrite(stdev_pval_fname, stdev_pval_temp);
%     ci_SNP_A = bootci(500,bootfun,rank_matrix(1:100,3));
%     ci_SNP_B = bootci(500,bootfun,rank_matrix(101:200,3));
%     ci_SNP_C = bootci(500,bootfun,rank_matrix(201:300,3));

    % concat data with all RR samples
    pval_all_rr_vals = cat(2,pval_all_rr_vals,average_pval_temp);
    stdev_pval_all_rr_vals = cat(2,stdev_pval_all_rr_vals,stdev_pval_temp);
    rank_matrix = [];
end % for param_1
final_rank_fname = sprintf('final_HAP_rank_ge.txt');
%eval(final_rank_fname);
dlmwrite(final_rank_fname,rank_all_rr_vals);
final_pval_fname = sprintf('final_HAP_pval_ge.txt');
%eval(final_pval_fname);
dlmwrite(final_pval_fname,pval_all_rr_vals);
final_stdev_pval_fname = sprintf('final_HAP_pval_stdev_ge.txt');
%eval(final_pval_fname);
dlmwrite(final_stdev_pval_fname,stdev_pval_all_rr_vals);
```

REFERENCES

1. Wang, W.Y., et al., *Genome-wide association studies: theoretical and practical concerns.* Nat Rev Genet, 2005. **6**(2): p. 109-18.

2. Couzin, J. and J. Kaiser, *GENOME-WIDE ASSOCIATION: Closing the Net on Common Disease Genes.* Science, 2007. **316**(5826): p. 820-822.

3. Blangero, J., *Localization and identification of human quantitative trait loci: King Harvest has surely come.* Current Opinion in Genetics & Development, 2004. **14**(3): p. 233-240.

4. Lohmueller, K.E., et al., *Meta-analysis of genetic association studies supports a contribution of common variants to susceptibility to common disease.* Nat Genet, 2003. **33**(2): p. 177-182.

5. Ioannidis, J.P.A., et al., *Genetic associations in large versus small studies: an empirical assessment.* The Lancet, 2003. **361**(9357): p. 567-571.

6. Pearson, J.V., et al., *Identification of the Genetic Basis for Complex Disorders by Use of Pooling-Based Genomewide Single-Nucleotide-Polymorphism Association Studies.* American Journal of Human Genetics, 2007. **80**(1): p. 126-139.

7. Craig, D.W., et al., *Identification of disease causing loci using an array-based genotyping approach on pooled DNA.* BMC Genomics, 2005. **6**: p. 138.

8. Butcher, L.M., et al., *SNPs, microarrays and pooled DNA: identification of four loci associated with mild mental impairment in a sample of 6000 children.* Hum Mol Genet, 2005. **14**(10): p. 1315-25.

9. Pritchard, J.K. and N.J. Cox, *The allelic architecture of human disease genes: common disease-common variant... or not?* Hum. Mol. Genet., 2002. **11**(20): p. 2417-2423.

10. Smith, D.J. and A.J. Lusis, *The allelic structure of common disease.* Hum. Mol. Genet., 2002. **11**(20): p. 2455-2461.

11. Risch, N., *The Genetic Epidemiology of Cancer: Interpreting Family and Twin Studies and Their Implications for Molecular Genetic Approaches.* Cancer Epidemiol Biomarkers Prev, 2001. **10**(7): p. 733-741.

12. Sladek, R., et al., *A genome-wide association study identifies novel risk loci for type 2 diabetes.* Nature, 2007. **445**(7130): p. 881-885.

13. Scott, L.J., et al., *A Genome-Wide Association Study of Type 2 Diabetes in Finns Detects Multiple Susceptibility Variants.* Science, 2007. **316**(5829): p. 1341-1345.

14. Diabetes Genetics Initiative of Broad Institute of Harvard and Mit, L.U.a.N.I.o.B.R., et al., *Genome-Wide Association Analysis Identifies Loci for Type 2 Diabetes and Triglyceride Levels.* Science, 2007. **316**(5829): p. 1331-1336.

15. Rich, S.S., *Mapping genes in diabetes. Genetic epidemiological perspective.* Diabetes, 1990. **39**(11): p. 1315-9.

16. Hirschhorn, J.N. and M.J. Daly, *Genome-wide association studies for common diseases and complex traits.* Nat Rev Genet, 2005. **6**(2): p. 95-108.

17. Hugot, J.P., *Association of NOD2 leucine-rich repeat variants with susceptibility to Crohn's disease.* Nature, 2001. **411**: p. 599-603.

18. Ogura, Y., *A frameshift mutation in NOD2 associated with susceptibility to Crohn's disease.* Nature, 2001. **411**: p. 603-606.

19. Rioux, J.D., *Genetic variation in the 5q31 cytokine gene cluster confers susceptibility to Crohn disease.* Nature Genet., 2001. **29**: p. 223-228.

20. Stoll, M., *Genetic variation in DLG5 is associated with inflammatory bowel disease.* Nature Genet., 2004. **36**: p. 476-480.

21. Daly, M.J. and J.D. Rioux, *New approaches to gene hunting in IBD.* Inflamm. Bowel Dis., 2004. **10**: p. 312-317.

22. Risch, N. and K. Merikangas, *The future of genetic studies of complex human diseases.* Science, 1996. **273**: p. 1516-1517.

23. Risch, N.J., *Searching for genetic determinants in the new millennium.* Nature, 2000. **405**: p. 847-856.

24. Cardon, L.R. and J.I. Bell, *Association study designs for complex diseases.* Nature Rev. Genet., 2001. **2**: p. 91-99.

25. Tabor, H.K., N.J. Risch, and R.M. Myers, *Candidate-gene approaches for studying complex genetic traits: practical considerations.* Nature Rev. Genet., 2002. **3**: p. 391-397.

26. *International HapMap Consortium. The International HapMap Project.* Nature, 2003. **426**: p. 789-796.

27. The International HapMap, C., *A haplotype map of the human genome.* Nature, 2005. **437**(7063): p. 1299-1320.

28. Pagani, F. and F.E. Baralle, *Genomic variants in exons and introns: identifying the splicing spoilers.* Nature Rev. Genet., 2004. **5**: p. 389-396.

29. Orr, H.A., *The population genetics of adaptation: the distribution of factors fixed during adaptive evolution.* Evolution, 1998. **52**: p. 935-949.

30. *International human genome sequencing consortium. Initial sequencing and analysis of the human genome.* Nature, 2001. **409**: p. 860-921.

31. Mira, M.T., *Susceptibility to leprosy is associated with PARK2 and PACRG.* Nature, 2004. **427**: p. 636-640.

32. Morley, M., *Genetic analysis of genome-wide variation in human gene expression.* Nature, 2004. **430**: p. 743-747.

33. Hoogendoorn, B., *Functional analysis of human promoter polymorphisms.* Hum. Mol. Genet., 2003. **12**: p. 2249-2254.

34. Lo, H.S., *Allelic variation in gene expression is common in the human genome.* Genome Res., 2003. **13**: p. 1855-1862.

35. *The International HapMap Project.* Nature, 2003. **426**: p. 789-796.

36. Bader, J.S., A. Bansal, and P.C. Sham, *Efficient SNP-based tests of association for quantitative phenotypes using pooled DNA.* GeneScreen, 2001. **1**: p. 143-150.

37. Jawaid, A., et al., *Optimal selection strategies for QTL mapping using pooled DNA samples.* Eur. J. Hum. Genet., 2002. **10**(2): p. 125-132.

38. Sham, P., et al., *DNA Pooling: a tool for large-scale association studies.* Nature Rev. Genet., 2002. **3**: p. 862-871.

39. Guohua Zou, H.Z., *The impacts of errors in individual genotyping and DNA pooling on association studies.* Genetic Epidemiology, 2004. **26**(1): p. 1-10.

40. Quade, S.R., R.C. Elston, and K.A. Goddard, *Estimating haplotype frequencies in pooled DNA samples when there is genotyping error.* BMC Genet, 2005. **6**(1): p. 25.

41. Gibbs, R.A., *The international HapMap project.* Nature, 2003. **426**: p. 789-796.

42. Luedeck, H. and R. Blascyk, *Fluorotyping of HLA-C: differential detection on amplicons by sequence-specific priming and fluorogenic probing.* Tissue Antigens, 1997. **50**: p. 627-638.

43. Le Hellard, S., *SNP genotyping on pooled DNAs: comparison of genotyping technologies and a semi automated method for data storage and analysis.* Nucleic Acids Res., 2002. **30**(15): p. e74.

44. Barratt, B.J., *Remapping the insulin gene/IDDM2 locus in type 1 diabetes.* Diabetes, 2004. **53**: p. 1884-1889.

45. Peter M. Visscher, S.L.H., *Simple method to analyze SNP-based association studies using DNA pools*. Gen. Epid., 2003. p. 291-296.

46. Scott, L.J., et al., *A Genome-Wide Association Study of Type 2 Diabetes in Finns Detects Multiple Susceptibility Variants*. Science, 2007. p. 1341-1345.

47. Zeggini, E., et al., *Replication of Genome-Wide Association Signals in UK Samples Reveals Risk Loci for Type 2 Diabetes.* Science, 2007. **316**(5829): p. 1336-1341.

48. Risch, N. and J. Teng, *The Relative Power of Family-Based and Case-Control Designs for Linkage Disequilibrium Studies of Complex Human Diseases I. DNA Pooling.* Genome Research, 1998. **8**(12): p. 1273-1288.

49. Dudek, S., Motsinger,AA., Velez,DR., Williams, SM., Ritchie, MD., *Data Simulation Software for Whole-Genome Association and Other Studies in Human Genetics.* Pacific Symposium on Biocomputing, 2006. **11**: p. 499-510.

50. Barrett, J.C., et al., *Haploview: analysis and visualization of LD and haplotype maps.* Bioinformatics, 2005. **21**(2): p. 263-265.

51. Yang, H.C., et al., *PDA: pooled DNA analyzer.* BMC Bioinformatics, 2006. **7**(1): p. 233.

52. Yang, Y., *Efficiency of single-nucleotide polymorphism haplotype estimation from pooled DNA.* Proc. Natl Acad. Sci. USA, 2003. **100**: p. 7225-7230.

53. Simpson, C.L., et al., *A central resource for accurate allele frequency estimation from pooled DNA genotyped on DNA microarrays.* Nucleic Acids Res, 2005. **33**(3): p. e25.

54. Hoogendoorn, B., *Cheap, accurate and rapid allele frequency estimation of single nucleotide polymorphisms by primer extension and DHPLC in DNA pools.* Hum. Genet., 2000. **107**: p. 488-493.

55. Kirov, G., et al., *Pooled DNA genotyping on Affymetrix SNP genotyping arrays.* BMC Genomics, 2006. **7**(1): p. 27.

56. Illumina, *Whole-Genome Genotyping with the Sentrix® HumanHap550 Genotyping BeadChip and the Infinium™II Assay.* 2006.

57. Grant, S.F.A., et al., *Variant of transcription factor 7-like 2 (TCF7L2) gene confers risk of type 2 diabetes.* Nat Genet, 2006. **38**(3): p. 320-323.

58. Hunt, K.J., et al., *Genome-Wide Linkage Analyses of Type 2 Diabetes in Mexican Americans: The San Antonio Family Diabetes/Gallbladder Study*. Diabetes, 2005. p. 2655-2662.

59. Florez, J.C., et al., *TCF7L2 Polymorphisms and Progression to Diabetes in the Diabetes Prevention Program.* NEJM, 2006. **355**(3): p. 241-250.

60. Groves, C.J., et al., *Association Analysis of 6,736 U.K. Subjects Provides Replication and Confirms TCF7L2 as a Type 2 Diabetes Susceptibility Gene With a Substantial Effect on Individual Risk.* Diabetes, 2006. **55**(9): p. 2640-2644.

61. Damcott, C.M., et al., *Polymorphisms in the Transcription Factor 7-Like 2 (TCF7L2) Gene Are Associated With Type 2 Diabetes in the Amish: Replication and Evidence for a Role in Both Insulin Secretion and Insulin Resistance.* Diabetes, 2006. **55**(9): p. 2654-2659.

62. Klein, R.J., et al., *Complement Factor H Polymorphism in Age-Related Macular Degeneration*. Science, 2005. p. 385-389.

63. Haines, J.L., et al., *Complement Factor H Variant Increases the Risk of Age-Related Macular Degeneration.* Science, 2005. **308**(5720): p. 419-421.

64. Thomas, D.C., *Are We Ready for Genome-wide Association Studies?* Cancer Epid. Biomarkers and Prev. , 2006. **15**(4): p. 595-598.

65. Pawitan, Y., et al., *False discovery rate, sensitivity and sample size for microarray studies*. Bioinformatics, 2005. p. 3017-3024.

66. Edwards, A.O., et al., *Complement Factor H Polymorphism and Age-Related Macular Degeneration.* Science, 2005. **308**(5720): p. 421-424.

67. Haines, J.L., et al., *Complement Factor H Variant Increases the Risk of Age-Related Macular Degeneration*. Science, 2005. p. 419-421.

68. Wacholder, S., *Assessing the probability that a positive report is false: an approach for molecular epidemiology studies.* J. Natl Cancer Inst., 2004. **96**: p. 434-442.

69. Carlson, C.S., *Selecting a maximally informative set of single-nucleotide polymorphisms for association analyses using linkage disequilibrium.* Am. J. Hum. Genet., 2004. **74**: p. 106-120.