A HAPLOTYPE-BASED PERMUTATION APPROACH IN GENE-BASED TESTING

by

Harrison Brand

B.S. Molecular, Cellular, and Developmental Biology, University of Michigan, 2004

M.P.H. in Epidemiology, University of Pittsburgh, 2009

Submitted to the Graduate Faculty of

the Department of Human Genetics in

the Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2013

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This dissertation was presented

by

Harrison Brand

It was defended on

January 29, 2013

and approved by

M. Michael Barmada, PhD, Associate Professor, Department of Human Genetics,
Graduate School of Public Health, University of Pittsburgh

Robert Ferrell, PhD, Professor, Department of Human Genetics,
Graduate School of Public Health, University of Pittsburgh

Dissertation Co-Advisor: Brenda Diergaarde, PhD, Assistant Professor, Department of Epidemiology
Graduate School of Public Health, University of Pittsburgh

Dissertation Co-Advisor: Eleanor Feingold, PhD, Professor, Department of Human Genetics,
Graduate School of Public Health, University of Pittsburgh

A HAPLOTYPE-BASED PERMUTATION APPROACH IN GENE-BASED TESTING

Harrison Brand, PhD

University of Pittsburgh, 2013

The soaring cost of health care is the biggest public health issue facing our country today. Development of strategies that improve the delivery of health care by identifying high risk individuals for a disease is a major approach to better utilize limited medical resources. Incorporating genomic data into risk stratification models is an essential component for creating these diagnostic and treatment strategies. Although initially applied to just small subsets of disease, advances in technology are making it economically feasible to utilize a patient's genomic data in a wider range of medical disorders. Current genetic association studies are crucial for identifying which loci to include in these models.

Genome Wide Association Studies (GWAS) are a valuable tool for identifying genetic variants associated with disease. Commonly, each SNP is initially independently tested in a GWAS with a univariate analysis. By combining the effects of multiple alleles, multivariate analysis of GWAS may increase power to detect associations and, thus, identify additional risk loci. We employ a haplotype block analysis within genes boundaries for a newly developed gene-based method, “GeneBlock”. GeneBlock is compared in a power analysis with two previously published permutation algorithms (GWiS and Fisher) and a simulation method (Vegas). All methods are tested in an Alzheimer Disease GWAS consisting of 1334 cases and 1475 controls. Results from the Alzheimer’s analysis were subsequently compared with haplotype and univariate analysis.

Power analyses shows both GeneBlock and GWiS as more powerful methods than Vegas and Fisher. A combinational approach involving the selection of the lowest p-value from Vegas, GWiS, and Geneblock has higher power than any individual method even when controlling for the additional multiple comparisons. Fisher and Vegas identify no significant genes in the Alzheimer's GWAS, while GWiS and Geneblock identified four (*PRDM16, ARHGEF16, HLA-DRA, TRAF1*) and three (*C17orf51, MGC29506, SLC23A1*) respectively. The combination method is also most powerful in the real GWAS data; it identified all seven of the above significant genes. Comparing single, haplotype, and gene level analyses revealed that only about 1/3 of the top 100 genes are shared, indicating a large variance in results between methods.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

PREFACE

I would like to thank several individuals who made this dissertation possible. Dr. Brenda Diergaarde has been a wonderful advisor for my five years at the School of Public Health. Her guidance helped me develop into the professional I am today. Dr. Eleanor Feingold has been critical in my growth as a statistician. Her assistance in developing the following power analysis was invaluable. Dr. Michael Baramada was amazing at helping me with any programming related questions. Dr. Robert Ferrell has been extremely valuable in providing a biological perspective of this dissertation. The Pittsburgh Center of Simulation and Modeling was pivotal in this project since they allowed me the use of their supercomputing resources. Specifically, Senthil Natarajan was amazing for his assistance on the PittGrid Supercomputer. Finally, I would like to thank Dr. Diergaarde, Dr. Joel Weissfield, and Dr. Mary Marazita for providing me with funding throughout my graduate career.

1.0 INTRODUCTION

Genome Wide Association Studies (GWAS) are a successful method for detecting genetic variation in complex diseases [McCarthy, et al. 2008]. Primarily, GWAS focus on the association between single nucleotide polymorphisms (SNPs) and a particular disease or trait. According to the National Human Genome Research Institute, around 1500 GWAS have been performed on a variety of different diseases and quantitative traits. Thousands of significantly associated variants have been identified in these studies. A complete catalog of published GWAS results is available at http://www.genome.gov/gwastudies/ (accessed 11/13/2012) [Hindorff, et al. 2009]. The majority of identified risk loci show only a small to moderate effect, which can be difficult to detect without large sample sizes. Many current studies underestimate the total number of SNP associations due to low power. The following dissertation presents a haploblock-based gene test (GeneBlock), which complements standard SNP analysis in which each SNP is tested independently to help identify risk variants previously unnoticed.

It was believed that GWAS would explain a larger portion of the genetic variation observed from familial studies. Unfortunately, the majority of heritability in common disorders remains hidden [Eichler, et al. 2010]. While it is likely that part of this unidentified heritability is located in other genetic variables such as rare variants, gene-environment interactions, and gene-gene interactions, GWAS data also still contains unrecognized risk loci [Manolio, et al. 2009]. Increased genome coverage has been suggested to improve identification of additional risk

markers. GWAS density is improved with either imputation or the increasingly large SNP arrays which contain millions of markers (HumanOmni5-Quad BeadChip, Illumina) [Marchini and Howie 2010]. Albeit there are benefits related to increased genome coverage, a significant problem arises when dealing with the many false positives associated with the millions of multiple comparisons. To help combat these false positives, a genome wide significant p-value threshold is often set around $5X10^{-8}$. Such a stringent cut-off greatly decreases the chance of false positives, but also prevents identification of many true positives. Several methods exist that can reduce the high dimensionality of a GWAS and therefore loosen the stringent significance cutoff. These methods generally combine SNPs into larger functional units of the genome such as haplotypes, genes, or even biological pathways.

1.1 FUNCTIONAL UNITS OF GWAS ANALYSIS

1.1.1 Haplotype Analysis

Haplotype analysis involves the investigation of a set of variants on the same transmitted chromosome. It cannot be directly carried out with GWAS data, since each SNP is genotyped independently, and it is unknown to what specific (father, mother) chromosome an allele belongs. Multiple algorithms exist to help alleviate this phasing issue [Browning and Browning 2011]. Rather than calculating a haplotype at a full chromosome level, it is much quicker to group SNPs into blocks of high linkage disequilibrium (LD) and phase within those blocks; a process known as haplotype blocking (haploblocking) [Gabriel, et al. 2002]. Therefore, GWAS data containing millions of markers at the SNP level can be reduced to hundreds of thousands of

haplotypes [He, et al. 2011]. Experimental evidence supports this notion; a recent GWAS by Lorenz et al. found that the use of haplotype blocking reduced the number of SNPs by approximately one-fifth and produced a stronger overall association in the highest observed risk loci [Lorenz, et al. 2010]. Besides reducing the number of multiple comparisons, haplotype analysis increases also the likelihood of identifying ungenotyped associated SNPs and can increase power to pick up multiple disease susceptibility alleles when they are in weak LD [Lorenz, et al. 2010; Morris and Kaplan 2002].

1.1.2 Gene Level

Biologically, genes are ideal to analyze as they specify protein structure and therefore have dramatic bearing on cellular processes. Mutations within coding regions or splice sites of a gene account for approximately 85% of known disease-causing mutations despite only accounting for ~1% of the total genome [Choi, et al. 2009]. It is no coincidence that genes are highly conserved across human populations [Neale and Sham 2004]. Gene-based testing has two clear advantages over both haplotype and single marker analysis; a dramatic reduction in multiple comparison due the relatively small number of genes (~20000) [Stein 2004], and increased power in situations where multiple markers are moderately associated with disease [Hibar, et al. 2011].

A major drawback of gene level analysis of GWAS data is the inability to analyze regions located in gene deserts. As a result, the method lacks the whole genome coverage available with SNP array data. These deserts are more important than once believed and even contain regulatory elements linked to neighboring genes [Ovcharenko, et al. 2005]. Numerous GWAS have implicated SNPs located outside genes as associated with a given disease [Grisanzio and Freedman 2010; Libioulle, et al. 2007]. Even though these regions likely have a

meaningful regulatory effect, interpretation of these peaks is difficult with current annotation information. In the future, a better understanding of these deserts may lead to similar multivariate analysis as gene-based testing within these areas.

1.1.3 Pathway Analysis

Pathway level analysis further reduces multiple comparisons with only around 420 annotated pathways in the popular KEGG database (http://www.genome.jp/kegg/docs/statistics.html, Accessed 10/27/2012) [Kanehisa and Goto 2000]. Numerous annotation issues currently still plague pathway analysis and, although future technology will likely help fill in missing gaps [Khatri, et al. 2012], for this reason we currently focus solely on gene-based methods.

1.2 GENE-BASED ANALYSIS METHODOLOGY

GWAS analysis with gene-based methods ideally would require a simple statistical test such as an omnibus regression to combine all polymorphisms within a gene. Unfortunately a standard regression model is often ineffective due to LD between SNPs, since it treats all SNPs as independent and therefore the degrees of freedom in the regression test will be over-estimated leading to a higher p-value. In order to help address this inherent LD, a variety of multivariate methods have been employed including principle component analysis [Gao, et al. 2011; Gauderman, et al. 2007], clustering based analysis [Buil, et al. 2009], Simes test [Li, et al. 2011], U-statistic [Li 2012], canonical correlation analysis [Tang and Ferreira 2012], and a multivariate Hotelling T^2 testing [Moskvina, et al. 2012].

Although each of the above noted statistical methods handles the lack of independence among markers, they come with assumptions of distribution in the data that often are difficult or impossible to verify. For example, the Hotelling T^2 assumes normality within its data [Mardia 1975]. Understanding the true distribution of genetic correlation within a gene has been attempted with the use of simulation, although it is difficult to elude [Li, et al. 2011]. Presumably, power to detect gene associations for different types LD structure may differ by gene-based method used. Therefore, it has been proposed that combining different statistical tests, such as a scaled chi-square and extended Simes' test, may improve results when applying the more powerful method under its optimal genetic structure [Bacanu 2012; Li, et al. 2012]. Under most conditions these hybrid approaches show greater power than their parent tests, but are still forced to predict an unknown distribution to derive a p-value. An alternative to standard parametric statistical tests is using non-parametric analyses, which are distribution free and therefore have no bias when handling various LD structures within a gene. Re-sampling based permutation testing is a popular non-parametric test since it is extremely flexible in a broad range of data [Motsinger-Reif 2008]. In gene-based testing, it is underutilized because it is extremely time consuming [Li, et al. 2011; Liu, et al. 2010].

1.2.1 Resampling-Based Permutation Testing

Resampling-based permutation testing is a powerful statistical test that does not require any assumptions of distribution since it is estimated through randomization of phenotype labels. By redistributing phenotype labels a distribution of observed test statistics is created under the null hypothesis of no expected differences between populations [Berger 2006]. An empirical p-value can be identified by dividing the number of times the replicate exceeds the true statistic divided

by the total number of replicates [Curtis, et al. 2008]. Permutation testing is extremely flexible and can be applied to any gene-based statistic or more generally to almost any statistical test. Furthermore, it only requires observations to be independent and identically distributed under the null hypothesis, both of which are generally accepted in genetic association studies [Phipson and Smyth 2010].

With the current explosive growth of computational power available to geneticists from next generation sequencing analysis, permutation testing for gene-based methods is becoming more of a realistic possibility at a genome-wide level. Key to this notion is the use of parallelization between multiple computers, allowing for nearly linear speeds of increased computing time. Software taking advantage of parallel computing is become commonplace in genetics [Steiss, et al. 2012; Zheng, et al. 2012]. Furthermore, other statistical methods are available to help accelerate these tests. In the vast majority of genetic studies involving hundreds to thousands of participants, it is impossible to enumerate all possible permutations. Monte Carlo Simulation addresses this issue by randomly selecting a subset of permutations while still maintaining similar power [Phipson and Smyth 2010].

1.2.1.1 Methods for Speeding Up Permutation Sequential Monte Carlo is an extension of a standard Monte Carlo simulation, which further decreases computational time by only requiring a set number of target test statistic a permuted data sets needs to exceed an observed test statistic [Besag and Clifford 1991; Curtis, et al. 2008]. Typically the targeted number of excessive replication is set to 10 which leads to an estimated 250-fold speed increase compared to regular Monte Carlo testing [Curtis, et al. 2008]. Huang et al. implement an additional step by estimating the number of causal SNPs with a Bayesian model selection for each gene and ignoring those without predicted risk variants. Therefore, permutations are only run on genes that have an

expected significant association [Huang, et al. 2011]. Liu et al. propose an alternative to permutation testing for gene level analysis, using a simulation to account for correlation between SNPs (see section 2.1.3 for more details) [Liu, et al. 2010]. In cases of a simple combinational statistic such as a chi-square test, the simulation method has similar results as a permutation test with a dramatic increase in speed. However, because p-values rather than actual genotypes are estimated with this method, its application to more complex multivariate methodology seems limited.

1.2.1.2 Issue with Permutation Testing In theory, permutation testing has the ability to control for LD between SNPs and therefore requires only simple methods for combining a univariate test statistic. [Curtis, et al. 2008]. However, whether LD structure is correctly accounted for solely with permutations remains controversial. A recent simulation by Moskvina et al. found a simple Fisher method-based approach failed to treat strongly related LD blocks as a single region and inflated the p-values [Moskvina, et al. 2012]. This could bias a genes' p-value in either direction depending on the location of the affected allele. If causative alleles were found in a tight LD block, p-values were over-estimated, whereas if non-significant p-values were found in a tight LD block, gene p-value tended to be under estimated. Moskvina et al. recommend the use of Hotelling's T^2 test instead which has a higher power in simulations and predicts more significantly associated genes than a Fisher-based permutation test. Interestingly, a study by Potter contradicts this finding and shows that Fisher-based permutation testing has comparable or greater power in most situations to detect true disease association compared to both Hotelling's T^2 and a U statistic [Potter 2006]. Considering the results from the Potter power analysis, we feel confident in permutation testing's ability to properly handle LD with the expectation that any weird LD structure will be identified in the follow up of significant genes.

1.2.3 Haplotype Analysis in Gene-Based Testing

Performing haplotype analyses in a gene-based setting was previously suggested for analysis of candidate gene association studies [Chapman, et al. 2003; Neale and Sham 2004]. TagSNPs were selected within haplotype blocks, and then haplotypes were phased with this information. A simple regression based analysis was carried out on the phased haplotypes to determine gene association [Neale and Sham 2004]. While these methods were effective at a candidate gene level, they are less appropriate in a GWAS setting since gene coverage varies and phasing is less effective when it must cover many SNPs in weak LD [Fallin and Schork 2000]. By implementing a blocking stage before phasing of SNPs, we will circumvent issues involving a large number of uncorrelated SNPs within a gene. A variety of methods exist to analyze haplotypes association within blocks with standard statistics such as chi-square or regression [Liu, et al. 2008]. The p-value obtained from one of these methods may then be treated identically to those from a single SNP analysis and combined using the Fisher method. Permutation testing should then be applied to account for the correlation between haplotype blocks (haploblocks), which are not completely independent and therefore violate the major assumption of the Fisher method.

1.3 CONCLUSION

I have implemented a haplotype-based permutation test within each gene for GWAS data analysis known as GeneBlock. GeneBlock is comprised of two steps. The first involves implementing a haplotype analysis, which can improve ability to find certain disease associated alleles compared to single SNP analysis [Lorenz, et al. 2010; Morris and Kaplan 2002]. The second step uses Resampling-Based permutation testing to control for correlation between haplotype blocks and derive and empirical p-value. This dissertation describes a comparison between the proposed GeneBlock method and two established gene-based tests that use permutation to account for LD (GWiS and Fisher) [Curtis, et al. 2008; Huang, et al. 2011]. In addition, I will compare GeneBlock with a variant of the Fisher method, which uses a computationally quicker simulation-based approach for controlling LD (Vegas) [Liu, et al. 2010]. Power and Type I error analyses are carried out with genes simulated to maintain their LD structure (Chapter 4). The simulations are then followed up by analyzing an Alzheimer's Disease (AD) GWAS data set using each method and comparing (Chapter 5).

2.0 GENE ANALYSIS METHODOLOGIES IN GWAS

Gene level analysis is a powerful tool that in conjunction with single SNP testing enhances GWAS results. A variety of gene level statistics exist, but since our approach focuses on permutation to control LD, we only discuss likewise methods (GWiS and Fisher) or those with a similar procedure (Vegas). The following section describes the methodology behind our proposed method, GeneBlock, and these other methods. A table highlighting each method is available at the end of the chapter (Table 1). Please note that terminology specified throughout the rest of this dissertation is specific for the case-control. Also, since most of the statistical analysis is carried out in the genetic analysis software Plink and unless otherwise specified please assume that testing occurred with the relevant Plink function [Purcell, et al. 2007b]

2.1 ESTABLISTED METHODS

2.1.1 Fisher

The Fisher method proposed by Curtis et al. is the simplest of all permutation methods [Curtis, et al. 2008]. P-values taken from a univiarite SNP analysis are combined using the Fisher method (equation 1) [Fisher 1925]. Fisher method derives a chi-square statistic by taking the $\log_e$ of the p-value and multiplying by -2. A combined p-value is then calculated from a chi-square distribution with df equal to 2 times the number of p-values.

Equation 1. Fisher Method

$$X^2 = -2\sum_{i=1}^{k} \log_e(p_i),$$

df=2k

Permutation testing is undertaken with the use of random phenotype labels. For each specific permutation a chi-square statistic is calculated using the Fisher method and then compared to the actual observed test statistic. Note, the Fisher method is equivalent to simply multiplying p-values due to the log multiplication rule and degrees of freedom are irrelevant since they stay constant throughout each replicate. As mentioned above, sequential Monte Carlo simulation drastically increases computation time and is therefore implemented in this Fisher approach [Besag and Clifford 1991]. Phenotype resampling stops when 10 permutated statistics exceed an actual test statistic. Once this threshold is achieved a p-value is obtained by dividing the 10 replicates (r) by the total permutations (n). In cases where r=0, an empirical p-value is derived from (r+1)/(n+1) so a p-value can never equal zero.

2.1.2 GWiS

The Gene-Wide Significant (GWiS) test combines independent effects within a gene with a greedy Bayesian model selection algorithm [Huang, et al. 2011]. The algorithm essentially runs the best tagging SNP in a regression model. SNPs are selected using a regression model with a greedy forward search, which sequentially selects variants that increase the Bayesian model likelihood of a model until all remaining SNPs only decrease the likelihood. Bayesian model selection picks the subset of SNPs which have the maximum model probability after correction for the number of parameters with a Schwariziain Bayesian Information Criterion. In the absence

of any association a null model is predicted which is automatically assigned a p-value of 1. The resulting test statistic is then permutated to find an empirical p-value.

2.1.3 Vegas

A versatile gene-based association study (VEGAS) is a unique method since it does not require genotype data, but rather works solely with individual marker p-values [Liu, et al. 2010]. P-values for each SNP in gene are converted to an upper tail chi-squared statistic and then added together to form a single gene chi-square statistics with one degree of freedom. LD between markers is accounted by using genotype data from a reference population. An empirical null distribution is created from a Monte Carlo simulation on a multivariate normally distributed random vector that has a correlation equal to those predicted from the references through a Cholesky decomposition matrix. A p-value is obtained as the proportion of simulated test statistics that exceed the observed gene-based test statistic. Vegas provides a pre-calculated LD matrix based on Hapmap populations (CEU, CHB, JPT, or YRI) [Thorisson, et al. 2005]. Reference populations can be assigned from user data sets also, but it is recommended to only use 200 randomly selected controls as a larger number of individuals require much more computational power with little effect on results.

2.2 GENEBLOCK

GeneBlock attempts to improve upon other gene-based test by the inclusion of haplotype blocks, which can identify new associations undistinguishable in other analysis. GeneBlock, similar to

the other gene-based permutation tests, has the additional advantage of operating with not only case-control data, but also familial or quantitative. Intriguingly, GeneBlock is likely more powerful within familial data than case-control, since haplotype phasing has better accuracy with known relatives [Marchini, et al. 2006]. An overview of the method is available in Figure 1 at the end of the chapter.

2.2.1 Blocking Methods

After SNPs are assigned to a relevant gene, they must be assigned to haploblocks. Many algorithms exist to classify these blocks including those based on information-theory [Anderson and Novembre 2003], LD [Gabriel, et al. 2002; Pattaro, et al. 2008], recombination [Wang, et al. 2002], and diversity of haplotypes [Patil, et al. 2001]. Numerous studies indicate that these methods can lead to dissimilar partitions, specifically the diversity-based approach by Patil et al. and the LD approach by Gabriel [Indap, et al. 2005; Pattaro, et al. 2008; Schwartz, et al. 2003]. The bulk of investigated algorithms in several independent studies seem to confirm the vast majority of blocking algorithms are similar to one of these methods [Bush, et al. 2009; Indap, et al. 2005; Pattaro, et al. 2008]. We therefore investigate both the LD approach (Gabriel) and diversity-based approach (HapBlock) to give an estimate of potential variation due to blocking method used. Blocks are assigned from both case and control data in order to prevent an additional time consuming step since blocks would have to be recalculated after each new random phenotype assignment if solely based on controls. In the power analysis described below this seemed to have little effect on SNP assignment to a specific block (results not shown).

2.2.1.1 Gabriel The Gabriel method assigns blocks with a confidence interval from pairwise D´ values [Gabriel, et al. 2002]. This method defines pairs of SNPs in strong LD if the one sided upper 95% confidence bound for D´ is greater than 0.98 and lower bound greater than 0.7. Anything with an upper bound D´ less than 0.9 is considered a weak LD region. Haplotype blocks are areas where less than 5% of comparisons show these weak regions. Overlapping blocks can be valid but generally the largest block is selected and then the biggest remaining blocks are included as long as they do not overlap with an already declared block. Gabriel et al. is the default method in Haploview and therefore widely implemented [Barrett 2009; Barrett, et al. 2005]. It is also available in Plink [Barrett, et al. 2005; Purcell, et al. 2007b].

2.2.1.2 HapBlock Patil et al. suggests a diversity method that designs blocks according to whether haplotypes found in at least 80% of the samples are represented more than once [Patil, et al. 2001]. Boundaries are then selected based on the maximum ratio of total SNPs compared to the minimal number of SNPs required to discriminate haplotypes within the block. Overlapping blocks are then removed and the process runs another iteration until the whole chromosome is covered. Unfortunately, the diversity-based method requires phased haplotype information to calculate blocks, which is unavailable in many genetic studies. Zhang et al extended this method to work for GWAS data by adding a haplotype inference algorithm that combines a haplotype-based dynamic programming algorithm with the partition-ligation-expectation-maximization algorithm to predict haplotypes [Qin, et al. 2002; Zhang, et al. 2002; Zhang, et al. 2004]. This expanded method can be found in the software HapBlock. [Zhang, et al. 2005].

2.2.2 Regression

Once the haploblocks within a gene are defined some form of haplotype association analysis must be carried out. Two main categories of methodology exist for analyzing case-control haplotype data; those comparing the frequency of a haplotype and those based on a regression framework [Liu, et al. 2008]. While both of these statistical tests are valid, we focus on regression analysis as it allows for the potential of controlling for covariates. Sham et al. proposes a haplotype regression based analysis which accounts for the potential ambiguity in haplotype phasing, a common criticism among similar methods [Purcell, et al. 2007a; Purcell, et al. 2007b; Sham, et al. 2004]. Furthermore, it is already implemented in Plink and works in sync with Plink derived blocking files.

The Sham et al. proposed haplotype analysis is essentially a single omnibus logistic regression test with one minus the number of haplotypes degrees of freedom (df). This jointly tests the effects of all haplotypes with an estimated frequency greater than 0.01. We selected a cut-off of 0.01, because that seems be the cut-off which anything below has dramatically lower haplotype phasing accuracy [Fallin and Schork 2000]. Haplotypes are predicted with an E-M algorithm, which gives estimated frequencies for each haplotype [Long, et al. 1995]. A posterior probability is then found for each haplotype with Bayes theorem and these are used as weights in a finite mixture regression model [Sham, et al. 2004]. This gives a likelihood function for each individual that can be compared with a Wald test to get a p-value [Purcell, et al. 2007b]. Those SNPs that are not assigned to a haploblock, are run with a standard logistic regression (0,1,2) with the common allele coded as 0.

2.2.3 P-value Combination

P-values derived from the regression analysis are subsequently combined using the Fisher method (equation 1) [Fisher 1925]. The Fisher method performed well in comparisons with other combinational methods such as the Tippett method [Tippett 1952], Liptak method [Liptak 1958], and Simes test [Simes 1986] in regards to permutation testing [Potter and Griffiths 2006]. Methodology of the Fisher Method can be found above in section 2.1.1.

2.2.4 Permutation

We follow the guidelines established by the Curtis et al. method using resampled case-control labels and described in detail in section 2.1.1 [Curtis, et al. 2008]. The empirical p-value obtained from this permutation maybe an under-estimate. Recall, a Monte Carlo simulation p-value is obtained by p=r/n, which r represents the permutated replicates which exceed the test statistic and n is the total permutations. Several papers suggest the true empirical p-value from is derived from (r+1)/(n+1) [North, et al. 2002; Phipson and Smyth 2010]. This has the advantage of never calling a p-value zero, which would be impossible if all possible permutations were calculated. Many statisticians however prefer the standard r/n derived p-value [Broman and Caffo 2003; Ewens 2003]. Note, the relative difference between these approaches is small and we therefore derive the p-value from r/n, as that is more prominent in existing methods [Huang, et al. 2011; Liu, et al. 2010]. In cases were r=0, we do use the corrected Monte Carlo p-value estimated with (r+1)/(n+1) since an empirical p-value should never equal zero [Phipson and Smyth 2010]. Total permutations are limited to 1,000,000 allowing the ability to determine significance at even a strict Bonferroni level (0.05/20000= 2.5e-06). In rare situations, the

haplotype regression will fail to converge for a certain replicate as the haplotypes in one group will become monomorphic and logistic regression is therefore unable to produce a p-value. These haplotypes are excluded from analysis.

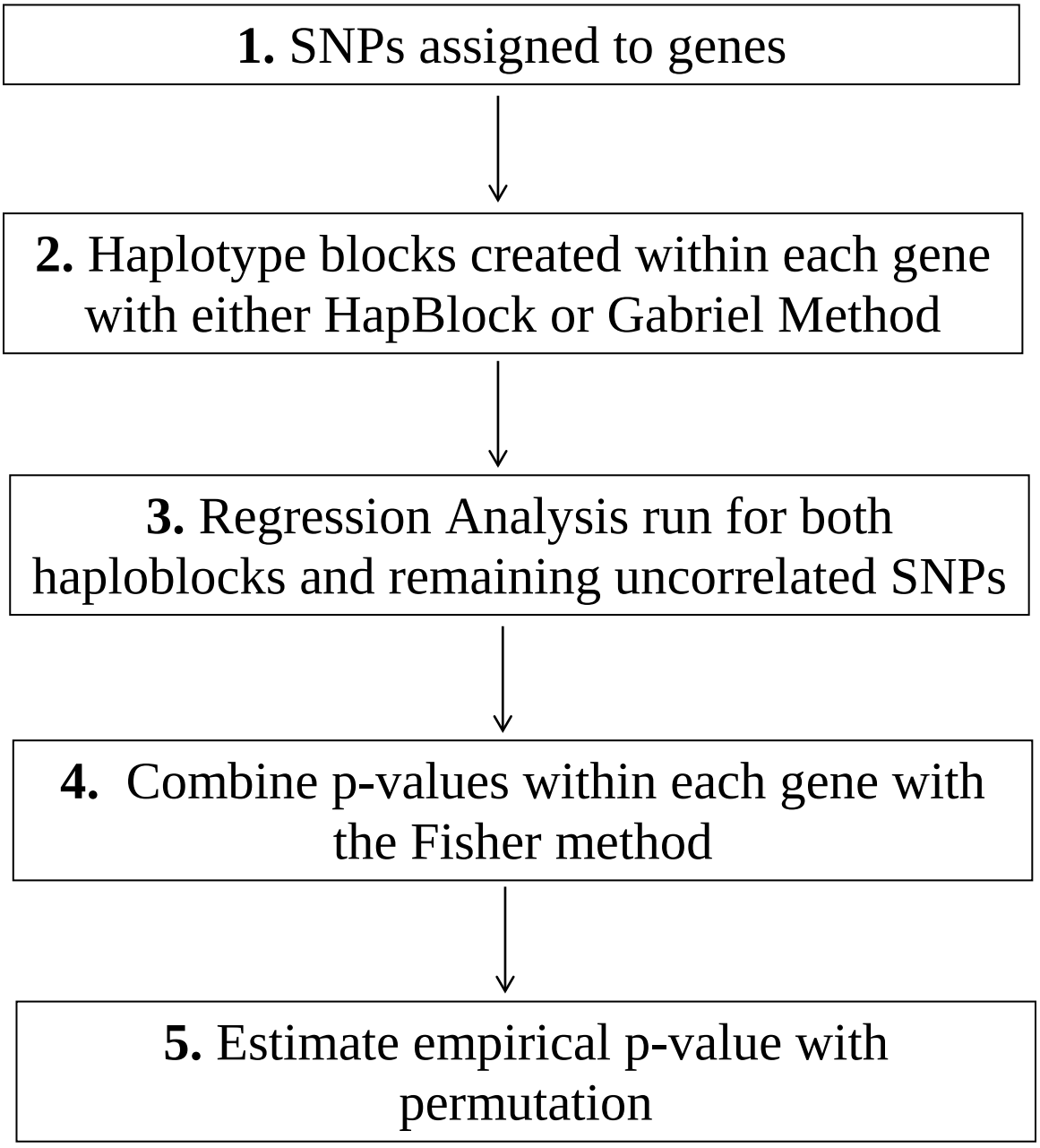

Figure 1. GeneBlock Algorithm

Table 1. Summary Info for Gene-Based Methods

	Vegas	Fisher	GeneBlock	GWiS
Combination Test Statistic	Chi-Square	Fisher Method	Fisher Method	Linear Regression
Method of Deriving P-value	Simulation	Permutation	Permutation	Permutation
Computational Speed*	Fast	Slow	Slow	Fast
LD Control	Simulation	Permutation	Haploblocks Permutation	Model Selection Permutation
Data Required	SNP	SNP	Haplotype#	SNP

*Direct analysis in Section 3.3

Can be estimated from SNP data

3.0 COMPUTATIONAL INFORMATION

3.1 COMPUTER HARDWARE

Permutation testing is computationally intensive and therefore I used supercomputing to alleviate this burden. Computer resources were provided by University of Pittsburgh Center for Simulation and Modeling (http://www.sam.pitt.edu/index.php, accessed 11/4/2012). Two supercomputing clusters (Frank, Pittgrid) were employed depending on the method. Frank is the more powerful of the two supercomputers, but is also in much higher demand and has fewer nodes. It is ideal for memory intensive programs. The Pittgrid can run hundreds of low memory nodes at the same time allowing for a large number of permutations to be carried out at once. For details on each system, see below.

3.1.1 Frank

The high-end computational system FRANK consists of 4352 CPU cores distributed over 325 nodes. It is run with a TORQUE Resource Manager [Staples 2006]. Frank has the ability to use up to 256 GB of RAM. As a non-investor (student) user, I can run approximately 48 cores at once on the shared memory partition. This could have been slightly increased if I ran on other memory partitions, but this was never necessary (http://core.sam.pitt.edu/frank, accessed 11/4/2012)

3.1.2 Pittgrid

Pittgrid combines the power of idle computers on campus by connecting them through a condor grid system [Thain, et al. 2005]. It contains 125 Linux and 300 windows computers. As this system requires idle computers, there is much more variability in its availability, since more people are using these computers during the day. Also there is a larger deviation in computing power between different processors in the computers found on the grid. Analysis time may therefore vary due to which computers are assigned to the project. A limit of 1000 jobs is recommended per submission. (http://www.pittgrid.pitt.edu/about.php,accessed 11/4/2012)

3.2 COMPUTER SOFTWARE

3.2.1 GeneBlock

The GeneBlock method with the Gabriel blocking approach (GeneBlock-Gabriel) was implemented in R [R Development Core Team 2012]. I coded the haplotype blocking and regression parts of the program for use with Plink. GeneBlock code was designed to work on the Pitt Grid condor system. Pittgrid is the better system for GeneBlock since it can run hundreds of low memory nodes at once, allowing for a large number of permutations to be carried out. Because different computers on the grid have different versions of R, care must be used when selecting functions from R packages. Desired functions from MADAM and CaTools packages were therefore directly implemented into the code rather than the standard practice of installing the full packages. See Appendix B for code.

GeneBlock-HapBlock code shared much similarity with the GeneBlock-Gabriel code except while the Gabriel blocking method is implemented directly in Plink, the diversity based blocking of HapBlock is available in a separate software package (HapBlock) [Zhang, et al. 2005]. Therefore, GeneBlock-HapBlock required a mixture of unix scripts, R code, and HapBlock code. See Appendix B for code. A minor bug exists in the HapBlock software that leads it to crash, preventing the determination haploblocks. Thankfully this only affected a few genes, which were unfortunately forced to be excluded from analysis. Attempts to fix the bug were unsuccessful and no apparent pattern links the genes. Multiple nodes were tested for each failed gene with no success, indicating the problem likely lies within the HapBlock code.

3.2.2 Fisher

No publicly available code existed for the Fisher method. Consequently, I wrote code for this algorithm in R. Within this code I still used Plink for single-SNP regression analysis because it is computationally faster than R. See Appendix B for code.

3.2.3 Vegas

Vegas is implemented in a software package available at (http://gump.qimr.edu.au/VEGAS/). This requires R with packages mvtnorm and corpcor and Plink 1.07 [Purcell, et al. 2007b]. Vegas is extremely memory intensive for larger genes and we therefore employ it on the Frank computing grid. We greatly reduced necessary memory requirements by only using 200 randomly selected controls to define LD structure as suggested by the authors [Liu, et al. 2010]. See Appendix B for code giving proper input format. Vegas generally worked with 4GB of

RAM, but this was inadequate for a few larger genes which required 8GB. The Vegas software will occasionally run the same gene test two times. I was unable to find the cause of the bug, but was able to rectify the problematic results by using only the first gene result of the pair.

3.2.4 GWiS

GWiS is implemented in the Linux based GWiS v1.1 software package (http://128.220.136.46/wiki/baderlab/index.php/GWiS, accessed 11/4/2012). In addition to implementing the GWiS algorithm, it also calculates three other methods (minSNP, minSNP-P, BIMBAM). As these methods have already been shown to be of lower power than GWIS, they were not used in my analysis [Huang, et al. 2011]. GWiS requires the GNU Scientific Library (GSL) (http://www.gnu.org/software/gsl/,accessed 11/4/2012). Most of the linux based pittgrid computers did not have GSL installed, so GWiS was run on Frank. See appendix for GWiS input settings.GWiS is more rigid than the other software packages since gene borders are hardcoded at 20kb and only one chromosome can be run per job submission. With parallel computing this is a not a problem as submissions are usually done at a chromosome level. The hg18 list was modified ± 5kb to account for the hard coded 20kb instead of the desired 5kb. A potentially serious bug exists in GWiS 1.1 when analyzing real GWAS data. GWiS does not handle missing data well and excludes any variable with a coded missing value (-9). Therefore, imputation of missing values is recommended for GWAS data sets before running GWiS.

3.3 COMPUTATIONAL SPEED

Computational speed of the different software packages is difficult to compare since many factors affect speed of analyses. Specifically the number of users present on a given supercomputer can greatly impact the number of available nodes and therefore the speed of each package. A direct assessment of computational time was made by investigating a single medium sized gene (*TFAP2E)* from a simulated population (section 4.2.1). Results from the comparison of CPU time are available in Table 2. GWiS is the fastest method for a low number of permutations, while Vegas is easily the quickest method for a large number. In GWAS analysis, Vegas is the fastest method due to the increased time GWiS requires for strongly associated genes that require many permutations (results not shown). GeneBlock is the slowest method since it is dramatically slower in situations requiring a large number of permutations. GeneBlock could be optimized for faster run time with more complex programming.

Table 2. Comparison of Computational Speed

Method	100 Permutations (min)	1000 Permutations	1,000,000 Permutations
GWiS	0:03	0:06	13:13
Fisher	0:36	5:13	25:58
GeneBlock-Gabriel	1:06	6:50	36:32
GeneBlock-HapBlock	1:13	7:58	37:33
Vegas	1:23*	1:23*	1:28

* Using the minimum number of Simulation allowed by Vegas of 100,000

4.0 TYPE I ERROR AND POWER ANALYSIS

4.1 BACKGROUND

Assessing the accuracy of each gene-based method's ability to detect true and false positives is necessary to determine the most proficient of the given approaches (GeneBlock, GWiS, Fisher, and Vegas). In gene-based testing, true and false positives are often assessed with a simulation calculating type I error and power. Type I error measures the rate of an erroneous rejection of a true null hypothesis leading to a false positive. In terms of gene-based testing, this would imply genes which are predicted to be significantly associated with a disease, even though no real association exists. Type I error is controlled for through permutation and therefore each method provides a corrected type I error. We confirm these gene-based methods capability of controlling for type I error in the simulation described below.

Power is the probability that a test will observe a difference when in truth there is one (true positive). It is defined as 1-type II error rate (false negative). Power analysis in gene-based testing, involves the simulation of a genes based on real data insuring realistic LD structure. These genes are assigned several "risk" SNPs. The "risk" SNPs strength of effect can be selected with two approaches. One assigns these as a percentage of the total disease variation explained by a specific gene, [Huang, et al. 2011; Tang and Ferreira 2012] while the other works with a simple risk ratio [Gao, et al. 2011; Li, et al. 2011]. Ensuring simulated risk SNPs are independent is crucial since two markers in high LD will provide no additional information if both are selected as risk alleles. Independence is determined from R^2 values [Gao, et al. 2011; Huang, et al. 2011] or haplotype blocks [Li, et al. 2011; Moskvina, et al. 2012]. Following this general

framework, studies also often attempt to control for additional confounding factors (gene size, risk effect, number of risk loci, and LD structure) by investigating multiples genes while varying a given factor [Huang, et al. 2011; Li, et al. 2011; Tang and Ferreira 2012]. Results interpreted from these analyses reveal how sensitive each method is to the desired factor. Typically simulation for power analysis in gene-based testing attempt to account for some of these parameters but no consensus exists for which specific confounders should be investigated.

4.1.1 Terminology

Because simulation and permutation testing have overlapping terminology, I briefly describe here the notation that will be used in the following sections. Each simulation has multiple genes, for which **data sets** are simulated. For example, simulations for *GGT1* could use 100 **data sets** with the same *GGT1* SNPs but having slightly different genotypes for the individuals in the population. When permutations are run, phenotype labels are randomly swapped within each data set. Each time this happens a **replicate** is produced.

4.1.2 Design

Designing a simulation to assess power in gene-based permutation tests has difficulties not apparent with other parametric gene-based approaches. The heavy computation requirements make it extremely demanding to calculate precise p-values in a large number of simulated genes across multiple datasets. Two components drive the high computational resources necessary for proper power analysis; investigation of many datasets and determining an exact p-value that achieves genome-wide significance. Both of these components are important to consider since

investigating in a large number of data sets controls for potential variation within data sets of a specific gene, while observing difference of gene-based test at the lower end of the p-value distribution provides a more interesting assessment since in real GWAS analysis those are the genes of relevance. We therefore spilt our power analysis into two separate tests, which become computationally feasible. The first simulation (cross-dataset) calculated power by classifying datasets within a specific gene as true positive when there empirical p-value is below a threshold of 0.01. The rate of true positives is equivalent to power since each data set is simulated to be "disease causing". Using a threshold of 0.01 allows a cap of 1000 replicates per simulated data set, thereby enabling the simulation of 1000 data sets. The second simulation (small p-value) focused on the comparative power of the different methods for specific datasets by calculating exact empirical p-value using up to 1,000,000 permutation replicates per data set. These p-values were much lower than the cross-dataset power analysis. We compared different gene-based methods power, by assessing which had the lowest overall p-value. Due to the large number of replicates, we only investigated 10 simulated data sets per gene. Additionally, a smaller power analysis similar to the above mention cross-data analysis was carried out within the high-replicate analysis using all the data-sets for every gene with a threshold of p-value=5.75e-5.

4.1.3 SNP Filter

SNPs in simulation were filtered to exclude those with minor allele frequency (MAF) < 0.05 and/or Hardy-Weinberg Equilibrium (HWE) < 0.0001. HWE exact test was used for calculations [Wigginton, et al. 2005]. We did consider including rarer alleles (MAF<0.05) since gene testing has seen a resurgence for its ability to handle these variants within whole exome data. [Derkach, et al. 2012; Madsen and Browning 2009]. While many of the multivariate tests used in GWAS

can be applied to rare variant data, the most powerful is based on the collapsing of low-frequency variants into a single variable. The collapsed variable can then be analyzed with a single univariate test. However, collapsing-based approaches remain controversial in GWAS. Simulation testing by Kinnamon et al. showed no increase in power when comparing pooling-based statistics to a single SNP analysis [Kinnamon, et al. 2012]. These findings contradict other studies which demonstrate standard techniques as underpowered in detecting rare associated markers and indicate a need of increased sample sizes for adequate detection relative to common variants [Asimit and Zeggini 2010; De La Vega, et al. 2011]. For this reason, we focus only on common variants though the collapsing-based approaches for rare variant analysis certainly warrant further inquiry.

4.1.4 Gene Annotation

We assign the same gene boundaries for all methods in the simulation to ensure proper comparability. Gene boundaries vary depending on annotation information, which is constantly being updated. Studies using boundaries from both hg18 and an updated hg19 are found in the literature and because these different releases have mismatched gene and SNP locations they can produce inconsistent results [Fujita, et al. 2011]. As the majority of SNP arrays still rely on hg18 annotation, we focused on this version (Illumina, San Diego, CA).

Genes were assigned with Plink's hg18-glist annotations (http://pngu.mgh.harvard.edu/~purcell/Plink/res.shtml,accessed 10/30/2012) [Purcell, et al. 2007b]. Boundaries in this file were determined from the hg18 build in the UCSC table browser for all RefSeq genes (http://genome.ucsc.edu/). Hg18-glist included overlapping genes with the expectation they would be recognized during analysis when deemed significant, although genes

with identical locations were removed from list. Isoforms of a gene were combined to maximize the length of the gene. A border of 25 kb both up and downstream of a gene ensured SNPs involved with transcription factors and enhancers were not missed. Less than 5% of quantitative trait loci for gene expression lie more than 20 kb upstream of the transcription start site. There is enrichment within 20 kb downstream of a gene as well [Veyrieras, et al. 2008]. Adding 5 extra kb, provides a buffer to ensure no additional relevant SNPs are ignored. In total, We only included autosomal genes in the gene-list due to the inherent difficulty of analyzing sex chromosomes with GWAS data [Clayton 2009]. 18,870 genes on the 22 autosomal chromosomes are included in our annotation list. In certain situations, SNPs from genes with a "–" character in the name are counted multiple times due to an inherent glitch in the unix grep command. As the hg18 list contained genes with the "-" character, several instances existed where genes had the incorrect number of SNPs. A simple work around of converting "–" to "_ " in the gene list fixed this bug.

4.2 CROSS-DATASET POWER ASSESSMENT

We compared the different gene-based methods with a cross-dataset power analysis based on a recent simulation carried out by Li et al. [Li, et al. 2011]. Briefly, in the Li et al study, SNPs were situated within LD blocks with three different linkage disequilibrium scenarios (moderate, strong, linkage equilibrium). Three different gene sizes were investigated (Genes with 3 SNPs, 10 SNPs and 30 SNPs), with an arbitrary risk effect of 1.14 assigned to an additive, and multiplicative models. A null model with no assigned risk effect tested type I error. Risk alleles were assigned based on gene size (1, 2, and 6, respectively). One thousand data sets were

generated per gene, which included 1500 cases and 1500 controls. Type I error and power were determined for a variety of different gene-based tests (Logistic Regression, a non-permutation Fisher test, Simes, Vegas) with each scenario. A gene was viewed as having a significant association if the p-value was less than 0.05.

We modify this power analysis in several ways. Further investigation of gene sizes in both 1000 genome project data and real Alzheimer's data revealed that gene sizes of 3, 10, and 30 SNPs are unrealistic, most genes will have far more SNPs on a typical GWAS chip. We therefore increased the number of SNPs analyzed in each gene size (see section 4.2.1 below). Determining independent markers for the disease allele simulation with haplotype blocks as in the Li et al. power analysis would not be appropriate for our specific power analysis since it could potentially favor GeneBlocks method of block-regression analysis. As many other simulations employ R^2 values to determine independence among SNPs, we use Haploview LD based tagger to select disease SNPs [Barrett 2009; Barrett, et al. 2005].

Several additional adjustments were made compared with the Li et al simulation in order to save computational resources. We reduced simulated cases and control to 1000 each allow for quicker analysis. A valid number in other gene-based testing studies [Li 2008]. Also, rather than assessing both a multiplicative and additive model we assigned risk variants solely assuming a multiplicative model since the power results were consistent with an additive model in the Li simulation [Li, et al. 2011]. We select three disease SNPs as causative, based on results from Huang et al. that predicted an average of approximately 3 independent disease alleles for significantly associated genes with various electrocardiography measures [Huang, et al. 2011]. In the Huang et al. study, gene size (#SNPs) had little effect on the predicted number of disease associated alleles, so it was therefore decided to keep the number of disease loci constant at 3

regardless of gene size. The overall goal in simulation these genes is to produce one that will mimic a true disease associated gene, which would solely be identified by gene-based methods. Therefore, rather than arbitrarily setting the strength of a risk allele, a more individualized approach was taken. In theory, risk effect per disease allele, could be estimated with a power estimating program such as CaTS [Skol, et al. 2006] All that CaTs requires to estimate genotype relative risk is a given sample size and SNP MAF. Unfortunately, when implemented into the simulation these relative risks often lead to both over and under estimates of association. Risk effects were therefore determined with a sensitivity analysis based on investigation of simulated p-values with the goal that a risk allele has a p-value around 1E-5. Our last major deviation from the Li et al simulation involves using a more stringent cut-off of 0.01 instead of 0.05 since it gave a larger range of powers.

4.2.1 Simulated Data Sets

Genotype data was simulated with Hapgen2, which allows simulation of disease SNPs within the same gene while controlling for LD.[Su, et al. 2011] Hapgen2 simulates genomic regions through resampling of haplotypes from a reference population accounting for fine-scale recombination rates across the region. Risk alleles in simulated cases are oversampled compared to controls leading to increased association with disease. A polymorphism template was provided by 1000 Genome Project (August 2009 CEU haplotypes) - NCBI Build 36 (dbSNP b126) obtained at the Impute v1 webpage (https://mathgen.stats.ox.ac.uk/impute/impute_v1.html). See Appendix for Hapgen2 input info.

As indicated above, genes of different size were tested. Unfortunately, a reasonable cut-off for gene size based on number of SNPs was unknown; therefore Chromosome 1 on the 1000

genome template (see above) was used to estimate reasonable gene sizes. Chromosome 1 was deemed to be sufficient for evaluating gene size but allowed for much quicker estimates compared with use of all chromosomes. The template contained 670,051 SNPs assigned to 1,946 genes (Figure 2). Based on Figure 2, it was determined that a small gene would be considered 1-300 SNPs, a medium gene 300-1000 SNPs, and a large gene 1000 or more SNPs. Within the small and medium gene sets, three genes we selected that were closest to the mean of the gene range for each size (150 SNPs, 650 SNPs). Large genes were determined to be too computationally intensive for the simulation and were therefore excluded.

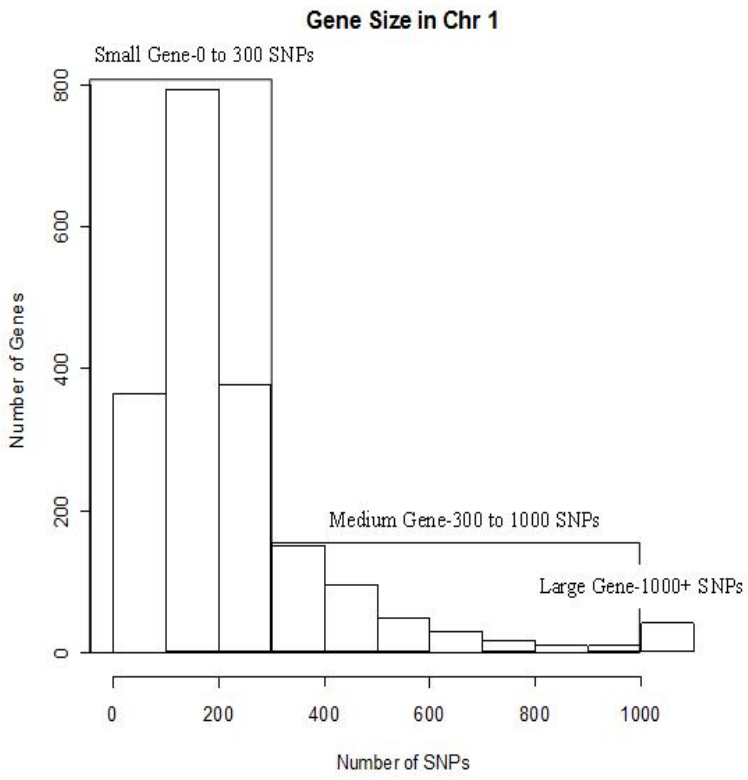

Figure 2. Gene Size (#SNPs) from Chromosome 1 of 1000 Genome Template

After the six genes were selected (3 small, 3 medium), three independent "disease" SNPs are chosen in each. Independence between SNPs was determined with haploview tagger, by selecting the minimal number of tagging SNPs in a gene given an r^2 threshold [Barrett 2009; de Bakker, et al. 2005]. In this case, we select an r^2 of 0.8 as our threshold for declaring independence as this is the haploview default. Also, since SNPs with a smaller MAF tend to have a large variation in risk effect during simulation, only SNPs with MAF >0.05 were considered for selection. SNP MAF varied with each data set so a "master set" was created for each gene consisting of 1,000,000 simulated controls. SNPs with a MAF <0.05 in the master set, were excluded as potential disease causing SNPs.

Data sets simulated for power analysis with Hapgen 2 consisted of 1000 cases and 1000 controls. Type I error was assessed under a null model which contained 2000 controls of which 1000 were randomly assigned as "cases." Under the null model no disease risk effects were simulated. As suggested in the SNP filtering section (4.1.3) data sets were filtered at a MAF 0.05 and HWE 0.0001. Data sets with any remaining SNPs with a p-value less than 1E-7 were resimulated as these would guarantee a significant gene p-value regardless of the method used to analyze the data. See Table 3 for a complete list of simulated genes with corresponding SNP information. Median p-values of all 1000 datasets in an example simulated gene (*SYPL2*) are presented in Figure 3. An outline of the simulation methodology is available in Figure 4.

Table 3. Simulated Genes for Large Sample Power Assessment

GENE	Size	SNP	BP	MAF	Risk Effect*
SYPL2	Small	rs6657193	109803625	0.09	1.6
		rs2787015	109810957	0.14	1.3
		rs41301283	109850980	0.47	1.3
FOXE3	Small	SNP_47637243	47637243	0.08	1.65
		rs72686267	47652169	0.09	1.6
		rs6697911	47680191	0.09	1.5
C1orf92	Small	rs4147301	155134879	0.09	1.6
		SNP_155145097	155145097	0.07	1.6
		rs17410711	155181833	0.1	1.6
TMEM51	Medium	rs72642314	15387636	0.13	1.5
		rs10927721	15407258	0.19	1.4
		rs12734436	15428203	0.23	1.3
HS2ST1	Medium	SNP_87226055	87226055	0.09	1.5
		SNP_87286338	87286338	0.14	1.5
		rs1419130	87349069	0.22	1.3
CACHD1	Medium	rs305550	64696235	0.07	1.6
		rs6686231	64814449	0.23	1.5
		rs12122465	64945896	0.15	1.5

*Effect size is multiplictive based on number of rare alleles

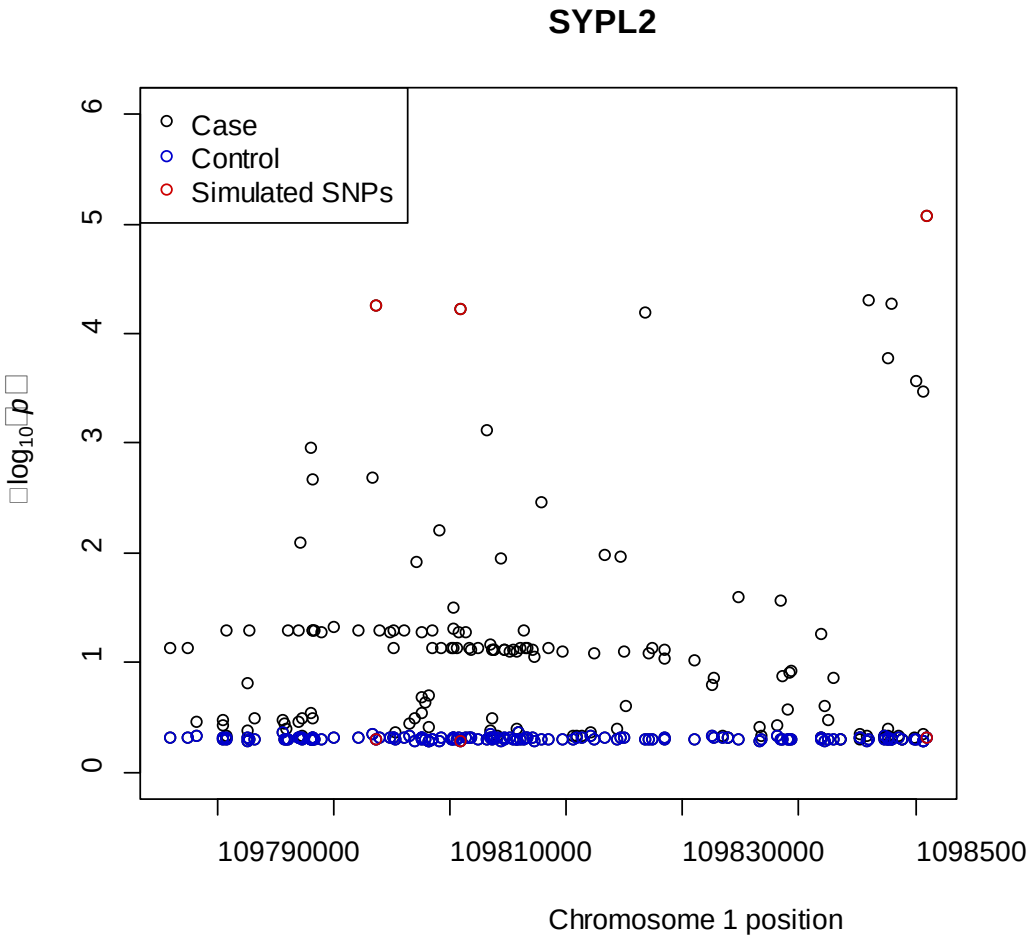

Figure 3. Example of Hapgen2 Simulated Gene with median P-Values

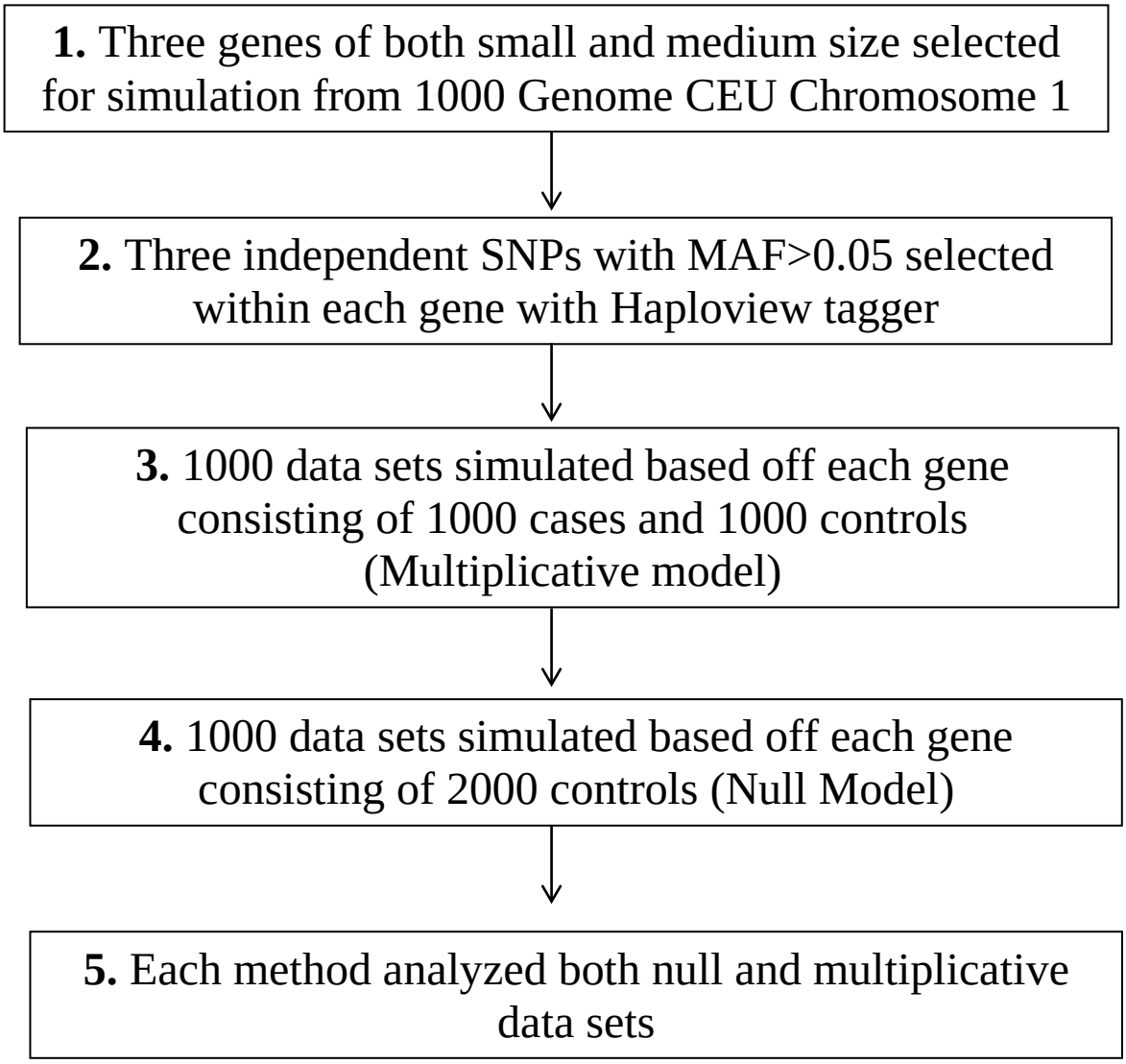

Figure 4. Simulation for Large Sample Power Assessment

4.2.2 Results

4.2.2.1 Necessity of Permutation In order to investigate the necessity of permutation testing, the GeneBlock-Gabriel method was analyzed with and without permutations. QQ plots were created from the simulated *SYPL2* null populations under both conditions. Figure 5 clearly indicates a wide deviation from the expected distribution for the uncorrected analysis. Interpreting these results using the standard p-values would permit many false positives, even when correcting for multiple comparisons. At an alpha of 0.05 we would expect 50/1000 simulated data set to have a p-value equal to or lower than 0.05. In reality, we find 143, almost three times the expected number. By adding permutation, we are able to correct these p-values and bring the type I error back to the expected level, with 53/1000 containing a p-value below 0.05 (Figure 6). By comparing the difference between expected false positives with and without permutation, the advantage of permutation testing is evident. Several published methods show QQ plots similar to the non-permutated GeneBlock method indicating the need for permutation testing to control for type I error with these statistics [Moskvina, et al. 2012; Tang and Ferreira 2012]

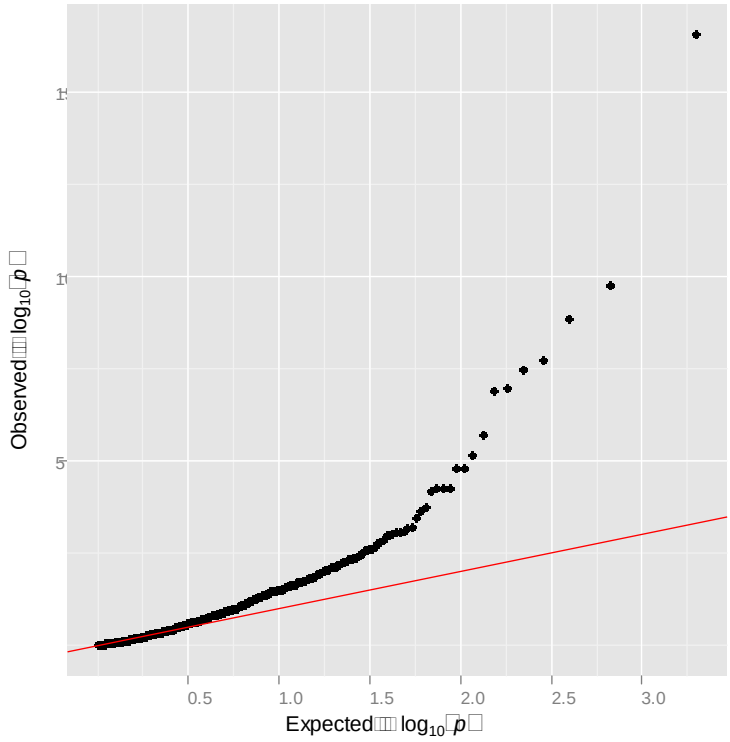

Figure 5. QQ Plot with Uncorrected GeneBlock –Gabriel P-Values

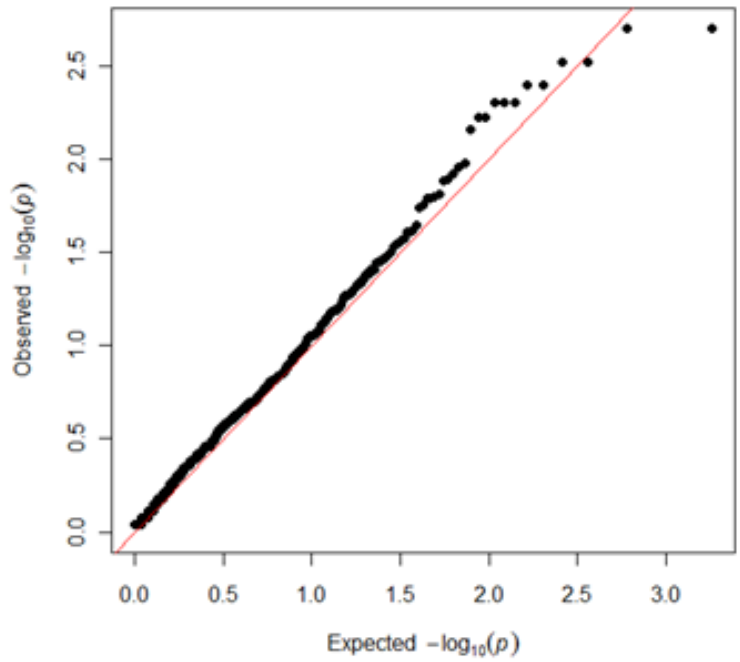

Figure 6. QQ Plot GeneBlock –Gabriel Permutated P-Values

4.2.2.2 Type I Error Type I Error, measured by the number of null data sets predicted as significant ($p<0.01$) are relatively equivalent for all the methods (Appendix Table A1). Type I error ranges from 0.005 to 0.016 giving only a 0.006 difference from the expected error rate of 0.01. We expected little variation in the Type I error because the p-values are corrected for with permutation.

4.2.2.3 Power Full results of the power analyses are available in Table A1 in the Appendix A. Figure 7 shows a comparison of power (number of data sets with p-value <0.01) for each gene. Vegas and the Fisher method have lower power than the other three methods. This is clearly more pronounced in the smaller genes, but is still apparent in some of the medium sized genes. Due to the similarity in the methodology between the Fisher and Vegas methods it is not surprising that they show such similarity in their results. Gabriel, HapBlock, and GWiS all have relatively similar powers throughout and each seems effective at detecting significant genes at a 0.01 level with powers generally greater than 90%. Fisher and Vegas power goes up dramatically in the medium sized genes (*CACHD*, *TMEM51*, *HS2ST1*) and are much closer to the power of the other methods. Therefore we can suggest Vegas and Fisher maybe underpowered to detect gene level association in smaller genes (1-300 SNPs), but further testing is certainly warranted. Note, 1 datasets in *CACHD1* had an error in block analysis with the HapBlock algorithm so they were automatically counted as false negatives (further details in section 3.2.1)

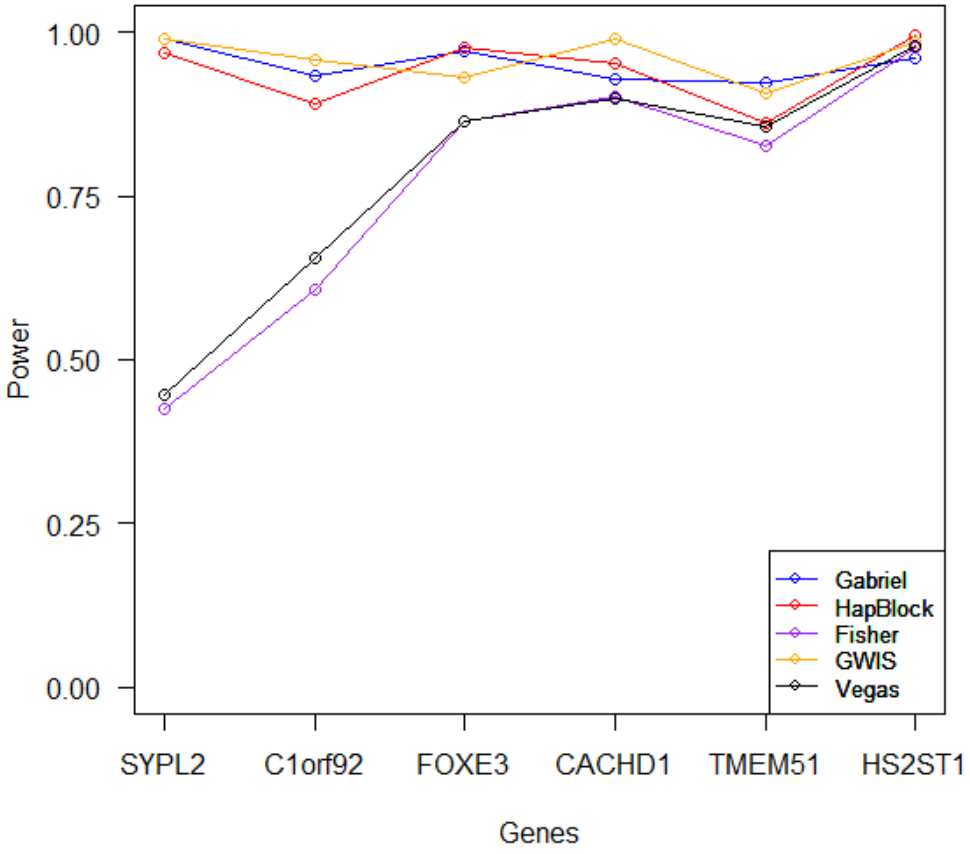

Figure 7. Power Analysis for Large Sample Power Assessment

4.3 SMALL P-PVALUE POWER ASSESSMENT

While the cross-dataset power analysis identifies variation between statistical tests at significance levels around p-value=0.01, it fails to gives us the more intriguing information with regards to how the methods power compare when detecting associations with extremely low p-values. These genes are of greater interest, because they have the potential to be statistically significant when accounting for the many multiple comparisons in a GWAS. In order to measure a minute p-value for a locus of interest permutations must increase dramatically from the 1000 seen in the cross-data set power analysis. A limit up to 1,000,000 permutations was set in this analysis, which has an ability to identify p-values greater than 1E-6. This dramatic increase in permutation forces us to focus on fewer data sets, so we reduce the total number of sets investigated per gene from the 1000 to 10 and focus on comparative power of the different methods for each individual dataset.

4.3.1 Simulated Data Sets

Genes were simulated with hapgen2 in the same manner as described above with a few notable exceptions. In the cross-dataset power analysis, each gene has 1000 data sets permutated 1000 times for a total of 1,000,000 replicates. The small p-value power analysis requires 1,000,000 permutations for 10 data sets for up to 10,000,000 replicates. Generating that many replicates with the above mentioned gene sizes, would be a huge resource burden. When selecting gene sizes, we had the choice of either estimating based on 1000 genome data, which will better represent more dense arrays, or we could estimate size based on the majority of available GWAS

data that are much less dense. For this specific analysis, it was decided to use the latter as the smaller gene sizes require a more reasonable amount of computing power.

If genes were randomly selected for analysis, it is likely only smaller genes would be analyzed; we therefore used genes of various SNP sizes to ensure proper applicability with a real GWAS. The relatively small number of data sets in each gene size (30), gene size though forced us to analyze the datasets in aggregate. A histogram of gene size (#SNPs) from the AD GWAS is shown in Figure 8. Based on the observed histogram, genes sizes were broken into three groups based on the number of SNPs (0-49, 50-150,>150). Data could not be simulated directly from the AD GWAS as hapgen2 requires haplotype and recombination information. Therefore, medians were determined for each range and genes with corresponding sizes were selected from a chromosome 1 of a 1000 genome template (described above). Small genes had a median number of 19 SNPs while medium and large had 71 and 207.5 respectively. Three genes were selected closest to each median number of SNPs. Selected genes: small (*AMY2A*, *HIST2H2AA3*, *MARCKSL1*), medium (*IVNS1ABP*, *LCE1A*, *TFAP2E*) and large (*GJA4*, *LAX1*, *SESN2*).

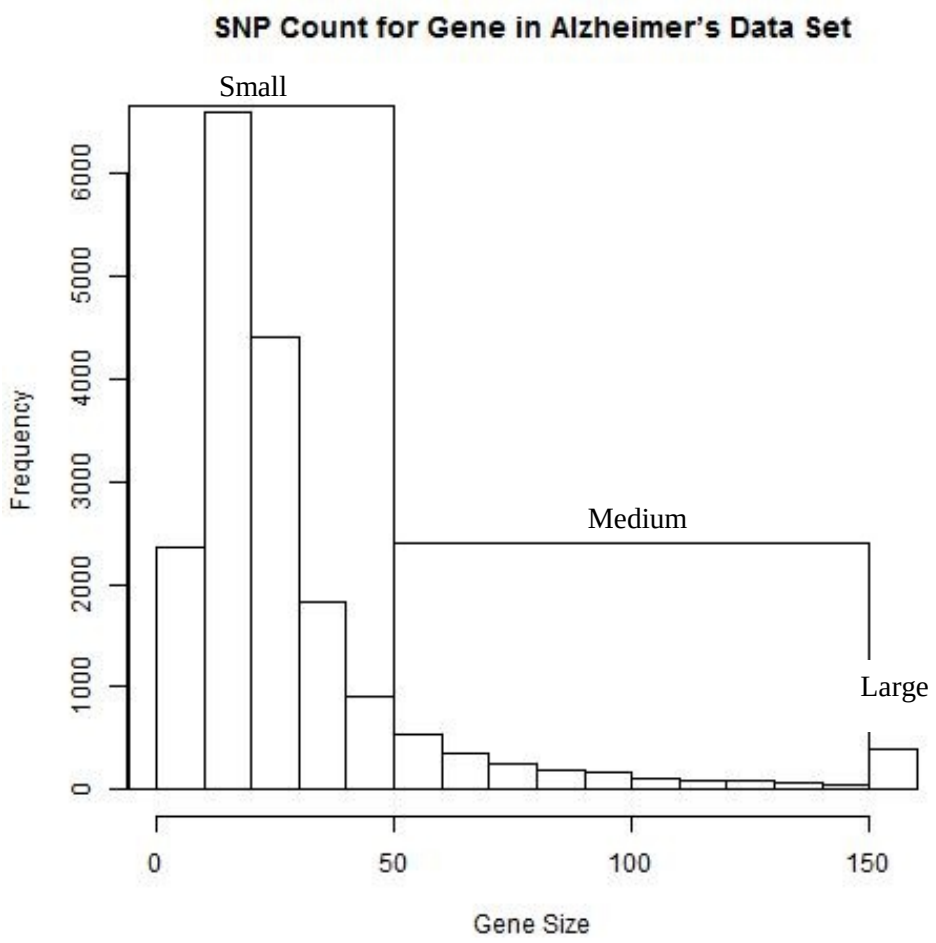

Figure 8. Gene Size (#SNPs) from Alzheimer GWAS

Due to the large variance in simulated risk allele effects, additional filtering was implemented. This was not possible in the cross-dataset analysis due to the large number of simulated data sets. Datasets that had risk variants with p-values between 5E-4 to 1E-6 were selected to ensure only reasonable risk effects drive gene p-values. Those with p-values beneath this threshold should already be identified in single-SNP analysis in GWAS negating the necessity of gene testing. Due to high LD, potentially a non-risk SNP could have p-value below 1E-6. Data sets with this situation were also excluded. Risk effects were estimated with a

sensitivity analysis as mention above (Section 4.2.1). See Table 4 for gene selection. Figure 9 provides a summary of the algorithm.

Table 4. Simulated Genes for Large Sample Power Assessment

Gene	Size	SNP	BP	MAF	Risk Effect
FAM72A	Small	rs1972478	204286061	0.0797	1.7
		SNP_204341537	204341537	0.0877	1.6
		SNP_204341722	204341722	0.21225	1.5
AMY2A	Small	SNP_103937608	103937608	0.0861	1.4
		SNP_103960308	103960308	0.1563	1.1
		SNP_103979216	103979216	0.1287	1.2
GJA4	Large	rs55757118	35011618	0.0487	1.4
		rs812239	35035363	0.14585	1.3
		SNP_35039105	35039105	0.14995	1.5
HIST2H2AA3	Small	rs56192814	148055552	0.05675	1.4
		SNP_148080377	148080377	0.06215	1.3
		SNP_148085185	148085185	0.0546	1.6
IVNS1ABP	Medium	rs1208517	1183539106	0.4497	1.1
		rs4651251	183564233	0.452	1.15
		SNP_183569003	183569003	0.11775	1.25
LAX1	Large	rs34568569	201987636	0.08915	1.4
		SNP_201987667	201987667	0.09255	1.4
		SNP_202030469	202030469	0.27165	1.2
LCE1A	Medium	rs12094590	151041957	0.09715	1.4
		rs35106590	151047045	0.0893	1.4
		SNP_151084390	151084390	0.19135	1.2
SESN2	Large	rs10494394	28446943	0.2198	1.25
		rs34315986	28458963	0.08585	1.3
		rs479144	28480925	0.0633	1.6
TFAP2E	Medium	rs12082263	35793880	0.0704	1.3
		rs6702475	35809319	0.12025	1.5
		SNP_35840445	35840445	0.1331	1.3

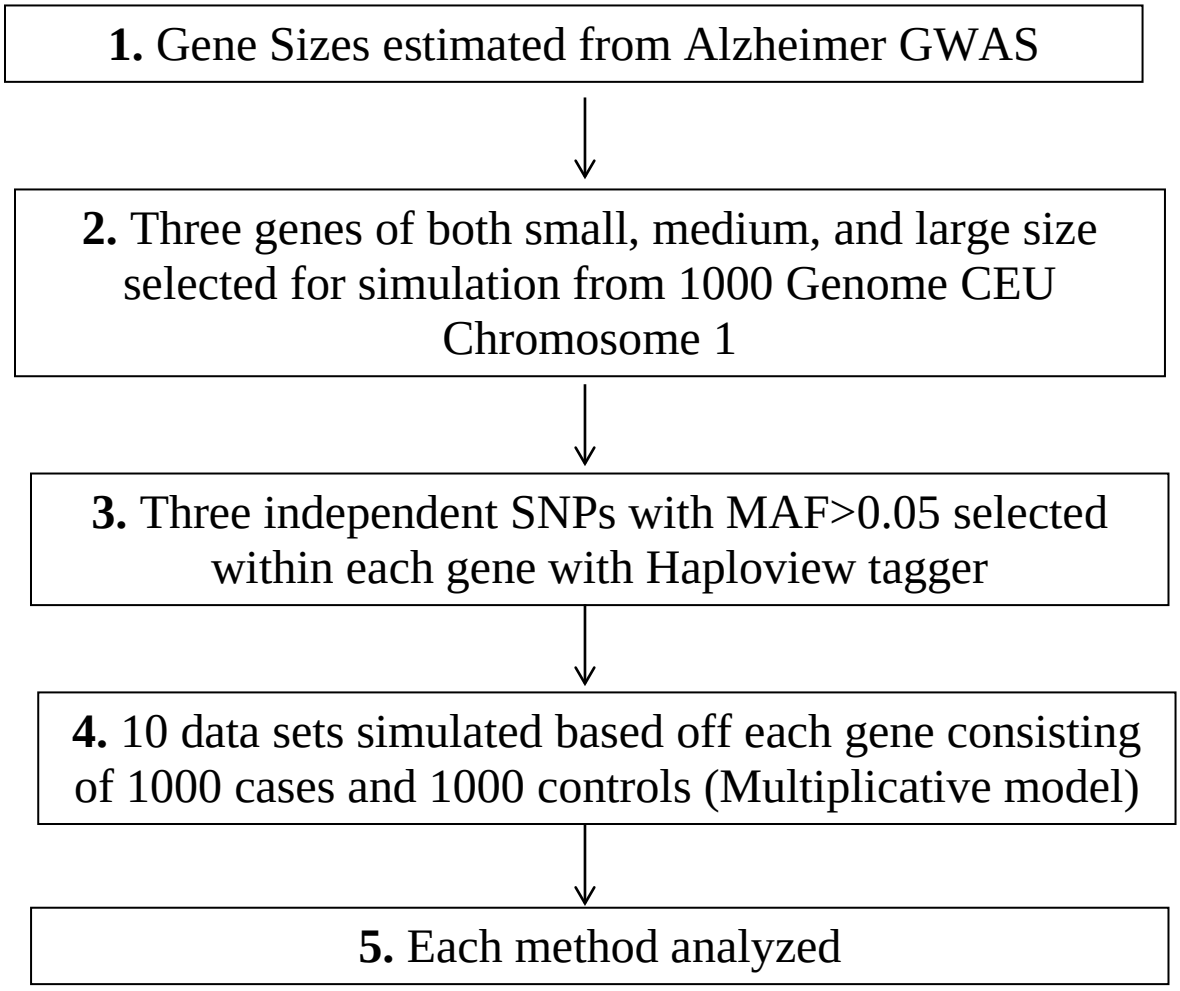

Figure 9. Simulation for Higher Replicate Power Assessment

4.3.2 Results

We performed a simple comparison of methods with the p-values for each simulated data set. Since all of the statistics have correct type I error (as ensured by permutation methods), the comparative power of the methods for a specific gene and dataset is directly indicated by the p-values. Results from this power analysis are in Table A2 in Appendix A. Investigating all 90 data sets (9 genes); the best overall method was GWiS, which has the most p-value approximately 36% of the time when averaging across all datasets in all genes. Both GeneBlock-Gabriel and GeneBlock-Hapblock have the lowest p-value around 23% of the time, while Vegas and the Fisher method are the top method about 10% of the time. Note, these percentages do not add to 100 because in some cases methods tied as most effective.

Figure 10 shows a comparison of -log10(p-value) means and 95% CI using each method for every gene. No method is significantly better than all other methods in any of the genes though several methods are stronger compared to other individual methods. A best method, which takes the lowest p-value from all methods, is included on this figure to show the potential of combining all approaches into a more powerful test. This should not be directly compared with the other methods as it involves all 450 p-values rather than 90 for each individual test. Therefore, it requires some sort of multiple comparison evaluation that would ideally be obtained from additional permutation testing. Results from this small-p value power analysis are in agreement with the cross-dataset power analysis in that GWiS, Geneblock-Gabriel, Geneblock-HapBlock are the top overall methods, with Vegas and Fisher showing substantially lower power.

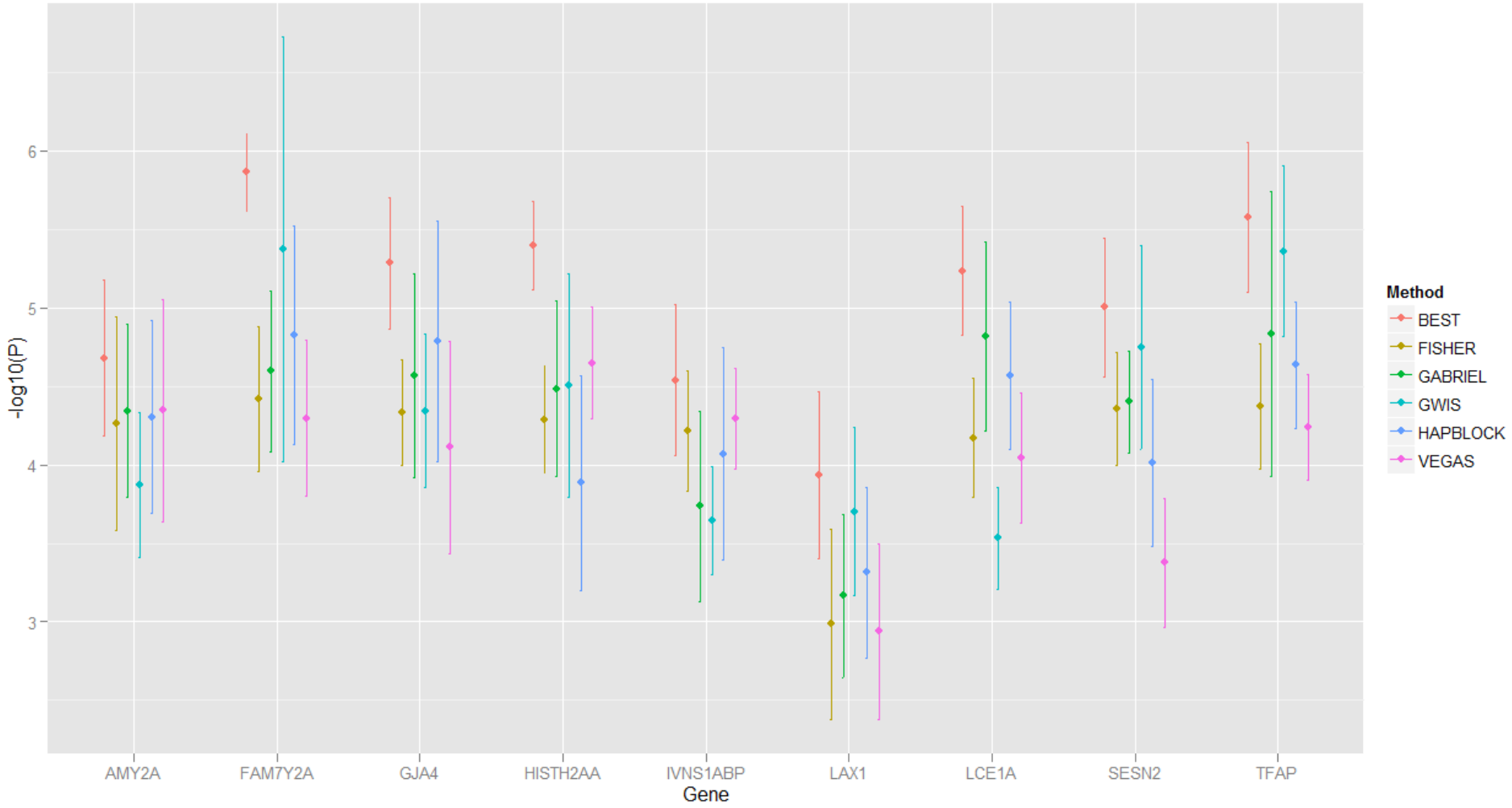

Figure 10. Average P-Value for Each Gene Based Method

4.3.2.1 Correlation We assessed correlation by combing all dataset across every gene and looking at the relation between the different methods (Figure 12). The top portion of the figure gives the correlation coefficient, while the bottom shows the correlation plot of all 90 points for each method. Fisher and Vegas are the most correlated with an r=0.67. This was expected due to the high methodical similarities between these techniques. Blocking methods are somewhat correlated with a r=0.47. GWiS is the least correlated with all other methods with $r \leq 0.3$. Certainly part of the lack of correlation is expected due to random variation from permutation testing. By comparing results from two separate Vegas runs on the same simulated data sets, we can measure how much of the missing correlation is due to chance. Figure 11 reveals an extremely high correlation with a r=0.99 indicating the lack of correlation is almost completely due to difference between the statistics and not the inherent variation in permutation testing.

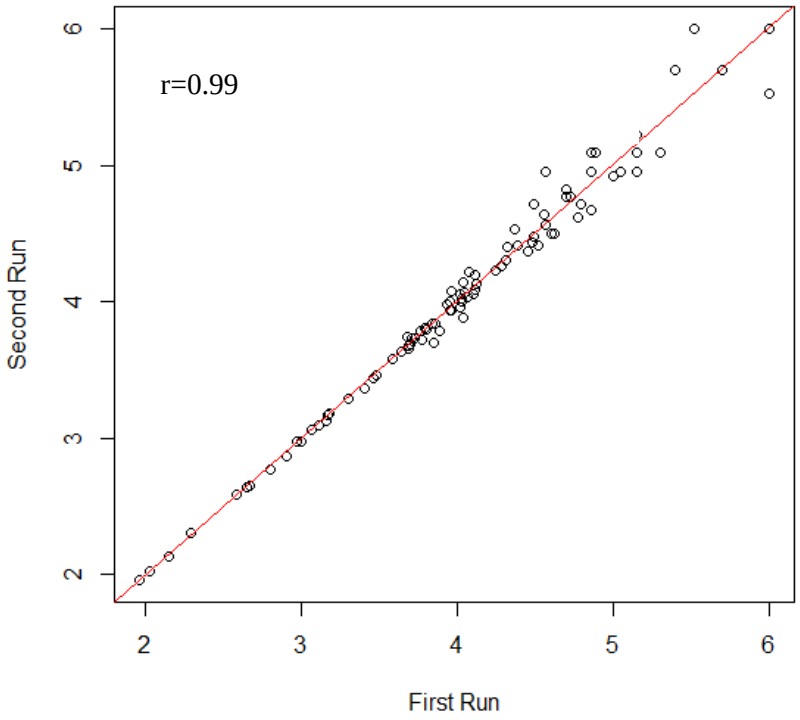

Figure 11. Correlation between Separate Vegas Simulations

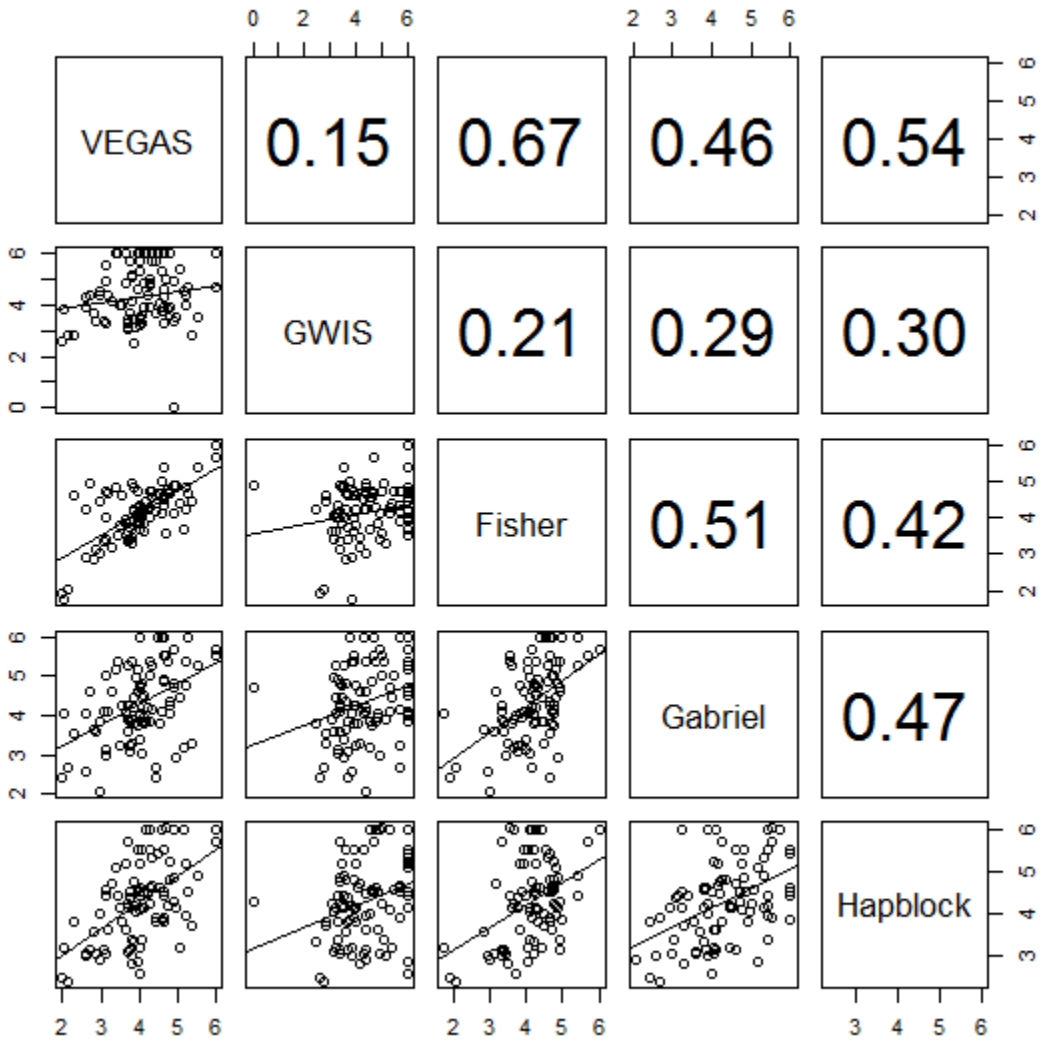

Figure 12. Correlation between Gene Methods

4.3.2.2 Comparison of Blocking Methods Block size was evaluated for each algorithm with both Gabreil and HapBlock method in all 90 data sets. Figure 13 and Figure 14 show an overlapping histogram of the two methods with all blocks and single SNPs (size 1). Histograms are splits apart in two sizes to get a better view of how blocks are assigned for each method. The Gabriel method produced blocks from 2-79 SNPs in length. HapBlock produced blocks of size 2-39 SNPs. Gabriel did not assign 653 SNPs to a haploblock, while HapBlock skipped 334. Hapblock has many more blocks sized 2-15 SNPs, while Gabriel has a higher ratio of blocks above 20. Previous studies have contradictory results with some showing the diversity method with bigger blocks [Pattaro, et al. 2008] and other agreeing with our results in which Gabriel are larger [Indap, et al. 2005; Schwartz, et al. 2003].

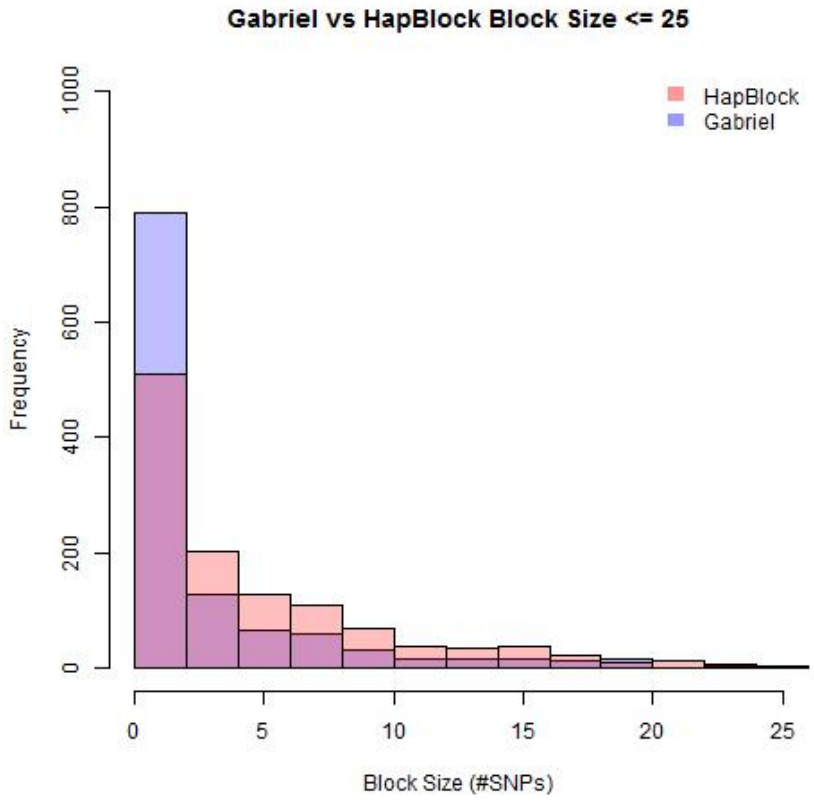

Figure 13 Gabriel vs HapBlock Block Size <=25

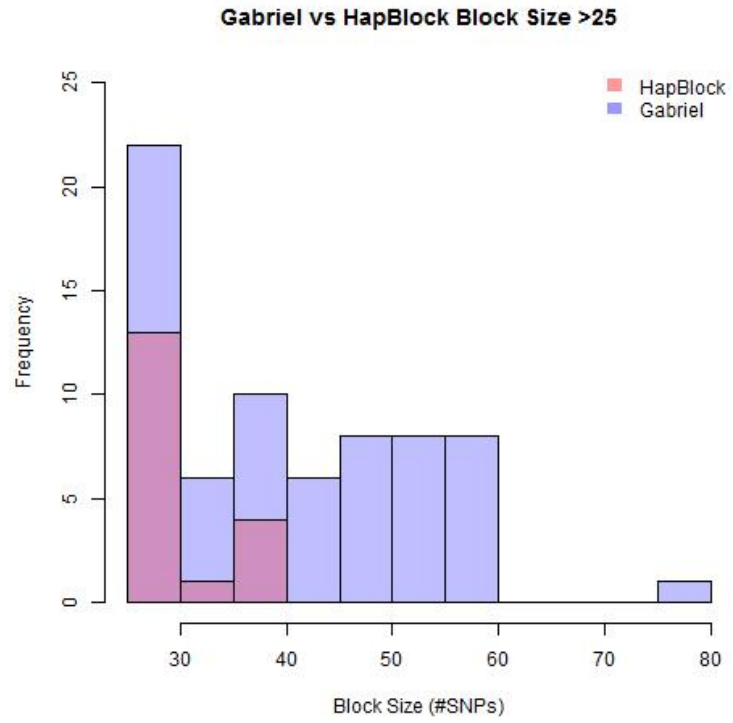

Figure 14. Gabriel vs HapBlock Block Size >25

The GeneBlock statistical test contains two blocking methods, which has the advantage of accounting for difference between blocking algorithms. It also has the disadvantage of doubling the number of statistical tests and therefore increasing the possibility of type I error. In order to determine the better blocking method, we investigated which the power analysis investigated with removing one. When the Gabriel method is ignored, the HapBlock method gives the lowest p-value 23/90 times. This was far below the top GWiS method, which is best 39/90 times. When the HapBlock method is ignored, Gabriel did extremely well by being the best method (33/90) and is nearly as strong as GWiS (36/90). Based on this simulation, we can assume that the Gabriel approach is the stronger of the two blocking methods.

4.3.2.3 Power Analysis A smaller power analysis similar to the cross-data analysis was conducted using a cut off of 1/17553=5.7E-5 estimated from the 17553 autosomal genes included in our AD GWAS. Since these are already corrected p-values anything below this value could be interpreted as significant even without a Bonferonni correction. Similar to the previous power analysis, both GeneBlock methods and GWiS preformed superior to Vegas and Fisher with a power of 50%. Fisher has similar power at 48%. Vegas is the weakest, with only a power of 39%. We further investigated the potential of a combination approach, which takes the lowest p-value from multiple gene-based methods. Taking into account the additional comparisons, we find a method combing the lowest p-value from Vegas, Gabriel, and GWiS to have a power that exceeds each individual test. This resulted in a power of 61%, which is likely underestimated due to the shared correlation among these genes and suggests the need for fewer multiple corrections. Vegas was selected over Fisher because they gave the same results and Vegas is much faster.

4.4 CONCLUSION

We performed a power analysis on four gene-based methods (Vegas, GeneBlock, GWiS, Fisher). Three of these (GeneBlock, GWiS, Fisher) control for LD between SNPs with random-label permutation, while Vegas uses an equivalent simulation-based method. The GeneBlock method used two blocking algorithms (Gabriel, HapBlock) for assigning haploblock within a gene. Permutation testing is time consuming and therefore power analysis with permutation-based methods requires a large amount of computational resources. In order to save computation time we devise two power analyses (cross-dataset, low-pvalue). Genes of multiple sizes were simulated for each power analysis with hapgen2 with parameters estimated from real data (gene size, risk effect, indepence among risk loci) or previously published studies (# risk loci, #genes investigated).

Cross-dataset power analysis found permutation adequately controlled for type I error in all gene-based methods. GeneBlock-Gabriel, GeneBlock-HapBlock, and GWiS all had relatively similar powers throughout that cross-dataset analysis, while Vegas and Fisher were significantly less, especially in smaller genes. The lowest p-value power analysis revealed GWiS as the top method 36% of the time when combing datasets from all genes. Both GeneBlock-Gabriel and GeneBlock-Hapblock had the next lowest p-values, 23% of the time, while Vegas and the Fisher where the best method 10% of the time. Further investigation of the low-pvalue analysis revealed Gabriel as the more effective blocking algorithm. According to the very specific genetic model investigated, which does not ever represent a fraction of the true models, GWiS is the most powerful gene-based method, though GeneBlock-Gabriel was almost equivalent.

5.0 GENE-BASED TESTING OF ALZHEIMER'S GWAS

Alzheimer's Disease (AD) is a neurodegenerative disorder common among the elderly. Prevalence in individuals 85 years of age and older is shockingly high with estimates ranging from 10% to 50% [Evans, et al. 1989; Fratiglioni, et al. 1999]. In 2012, approximately 5.2 million Americans are afflicted with this disorder [Hebert, et al. 2003]. With an aging population this number will grow dramatically over the next few decades as 13.2 Americans million are predicted to suffer from AD by 2050 [Hebert, et al. 2003]. Globally, greater than 24 million people suffer from AD with an expected increase to 81.1 million by 2040 [Ferri, et al. 2005].

The exact pathogenesis of AD is unknown but, extensive evidence supports a strong role for amyloid beta depositions in neuronal dysfunction [Ballard, et al. 2011]. A major protein involved in the formation of these amyloid plaques is APOE, which converts the monomeric form of amyloid beta into a more toxic oligomer or fibril conformation [Verghese, et al. 2011]. Early GWAS of AD identified extremely high peaks within SNPs on chromosome 19 in *APOE* [Bertram and Tanzi 2009]. The *APOE* association is widely replicated in numerous GWAS in a variety of different ethnic populations [Bertram and Tanzi 2009; Chung, et al. 2012; Logue, et al. 2011]. In addition, the *APOE* region is strongly associated with age-at-onset in AD [Coon, et al. 2007; Kamboh, et al. 2011]. Evidence of a role for genetic factors in AD is strong, with twin studies estimating heritability at 0.74, but *APOE* is only predicted to explain approximately 20% of this risk.[Gatz, et al. 1997; Slooter, et al. 1998]. Weaker risk variants have been identified but replication has not always been consistent [Ballard, et al. 2011; Bertram and Tanzi 2009]. Additional contributing genetic variants are likely hidden in these AD GWAS, but are difficult to find with such a stringent genome-wide significance driven by the many SNPs tested. Gene-

based testing is an effective method to reduce multiple comparisons with GWAS data and therefore may find previously unknown risk loci [Liu, et al. 2010].

Gene level analysis are underutilized in current AD research, with only a handful of studies identified [Hibar, et al. 2011; Li, et al. 2008; Velez, et al. 2012]. Instead, pathway-based analyses of GWAS data are the preferred approach for handling multiple comparisons [Hong, et al. 2010; Lambert, et al. 2010]. As mentioned in the introduction, these pathway-based approaches have the disadvantage of relying on inadequate annotation data [Khatri, et al. 2012]. In this section, we employ Vegas, GWIS, Fisher, and both GeneBlock methods to existing GWAS data in order to identify additional risk loci for AD. Results are compared with a standard single SNP and haplotype based analysis to demonstrate the effectiveness of these approaches.

5.1 MATERIALS AND METHODS

5.1.1 Study Population

Subjects for the AD GWAS were obtained from two different sources: the University Pittsburgh Alzheimer Disease Research Center (ADRC) [Kamboh, et al. 2012a; Kamboh, et al. 2012b] and an ulcerative colitis and Crohn's disease cohort [Achkar, et al. 2012]. The ADRC study population consists of 1440 Caucasian AD cases and 1000 Caucasian controls. All cases met the National Institute of Neurological Disorders and Stroke (NINCDS) and Alzheimer Disease and Related Disorders Association, Inc (ADRDA) criteria for probable or definite AD. Age and gender matched controls were obtained from the same region as cases, and had no psychiatric or neurological disorders including mild cognitive impairment or dementia. All participants gave

informed consent and the study received approval from University of Pittsburgh Institutional Review Board (IRB). An additional 508 Caucasian controls were included from the ulcerative colitis and Crohn's disease cohort from Ackbar et al [Achkar, et al. 2012]. Participants for this cohort were recruited from either Cleveland Clinic or the University of Pittsburgh. Subject recruitment followed each institutes IRB protocol and written informed consent was obtained from all participants.

Table 5 displays demographic information from each group. Only participants with high quality DNA (>98% of SNPs called) were included in the table (1334 cases,1475 controls) (see section 5.1.3.1). Covariate information was unavailable for 64 participants (43 cases, 21 controls) of those with high quality DNA. The mean age at enrollment is significantly lower than the mean age at enrollment of the ADRC control group (45.8 years versus 75.5 years; 2 sided t-test p-value<0.00001). This is most likely due to the fact that Achkar et al. enrolled controls age 20 and older while the ADRC enrolled controls age 58 and older. Since approximately 4% of AD is observed in people below 65 and 6% in people 65-74, a combination of these controls regardless of age seemed reasonable, because the vast majority of cases are 75 and older [AlzheimerAssociation 2012]. The difficulty in collecting healthy controls over 75 forces the collection of younger controls, some of whom will be diagnosed with AD later in life. Having some potential cases misclassified as controls will weaken the appearance of truly associated disease variants in analysis, but with the large sample size of this study this effect should be negligible. Gender is also significantly different between control groups with ADRC having 63.3% women and Ackhar et al. only containing 55.6% (chi-square p-value=0.004).

Table 5. Characteristics of the Study Population

	ADRC		Achkar et al.
	Cases (N_{total}=1291)	**Controls (N_{total}=958)**	**Controls (N_{total}=495)**
Age (in years; mean ± sd)	77.3 ± 6.3	75.5 ± 6.3	45.8 ± 13.6
Age (median)	77	76	45
Age Range	59-99	58-97	20-99
Sex [women; *N* (%)]	813 (63.0)	606 (63.3)	275 (55.6)
Mean Age at Onset (N)	72.8 ± 6.5 (1190)		
Median Age at Onset	73		

5.1.2 Genotyping

Genotyping was performed using the Illumina Omni1-Quad chip (San Diego, CA), which contains probes for in total 1,016,423 SNPs and/or copy-number variations. All samples were genotyped at the Feinstein Institute for Medical Research of the North Shore-Long Island Jewish Health System (Manhasset, NY)

5.1.3 Quality Control

5.1.3.1 Sample A total of 106 cases and 34 controls were excluded from analysis due to a genotypic failure rate above 2%. We included all remaining individuals including those with missing variables (age, gender) since no covariates were assessed. Therefore 1334 cases and 1475 controls were included in the analysis.

5.1.3.2 SNP Testing for Hardy-Weinberg equilibrium using the exact test removed 2,239 SNPs with significant deviations from expectation (P≤1E-06) [Wigginton, et al. 2005]. We observed

genotypic failure greater than 2% in 22,385 SNP, which subsequently were excluded. SNPs with a minor allele frequency < 5% were also removed (N=269,652) leaving a total of 723,397 SNPs. In addition, 16,047 non-autosomal SNPs and 15 duplicate SNPs that mapped to the same location as another SNP were excluded. The duplicate SNPs generally had identical genotypes to those with a shared location. A total of 707,335 markers were included in the final analysis with 405,444 SNPs located in or ± 25 kb of a gene in the Plink hg18-list (section 4.1.4).

5.1.4 Population Stratification

Population stratification analysis was performed in a previous study with the ADRC population using multi-dimensional scaling-based (MDS) methods available in Plink [Kamboh, et al. 2012b]. The GWiS software does not allow for covariates and thus we could not make use of the MDS population stratification variables. Population stratification by dividing with the genomic control (λ) is possible for single-SNP and gene-based method Vegas analysis, but would be ineffective for other gene level analysis [Price, et al. 2010]. We therefore ignored population stratification in the current analyses in order to allow a comparison of all methods. This will have little effect on the overall rank of the most highly associated SNPs and genes, but it will lead to over-estimated p-values at a genome-wide level in these analyses.

5.1.5 Imputation

The software package GWiS does not tolerate missing SNP data. Therefore we imputed all SNPs with a failed genotype call in Plink using control samples as a reference. Default settings from Plink were implemented (Appendix B). Plink imputation works through the phasing of proxy

SNPs in a reference panel and then calling of the most likely genotype. Proxy SNPs are selected based on LD with the unknown marker. Initially, 2,762,312 (0.1%) genotypes were categorized as missing. After imputation only 6272 (3e-4%) SNP genotypes remained undistinguishable. These were assigned as the common allele.

5.1.6 Analysis

5.1.6.1 Standard Single-SNP GWAS Analysis All 707,335 SNPs in the 2809 participants passing quality control were included in the AD GWAS analysis. We tested individual marker associations with logistic regression under an additive model (0,1,2) corresponding to the number of minor alleles. Manhattan and QQ-plots were created using R code from the Getting Genetics Done Blog. (http://gettinggeneticsdone.blogspot.com/2011/04/annotated-manhattan-plots-and-qq-plots.html, accessed 12/2/2012). Direct estimation of genome wide significance with a Bonferroni correction using all 707,335 SNPs is extremely conserved because many of the 707,335 SNPs are correlated and therefore should not be considered independent [Bonferroni 1935]. An effective number of independent markers was estimated through haplotype blocks as suggested by Duggal et al. and used to calculate a more accurate genome wide significance with a Bonferroni correction at an of $\alpha = 0.05$ [Duggal, et al. 2008]. Haploblocks were created across the genome with the Gabriel method (section 2.2.1.1).

5.1.6.2 Haplotype Analysis We perform a haplotype analysis on the AD GWAS data. Each of the 707,335 SNPs were assigned to haplotype blocks (haploblocks) created with the Gabriel algorithm as described in section 2.1.3.1. Haplotype regression within blocks was carried out in Plink using the Sham et al. method illustrated in section 2.1.4. Permutation testing within blocks

has been suggested to correct haploblock p-values[Purcell, et al. 2007b], but was avoided in this situation due the larger number of blocks and the high number of iterations required for smaller p-values. A multiple comparison correction for haploblocks was designated in the same manner as the single SNP analysis. Theoretically, haploblock analysis is performed after a single SNP analysis and therefore it is redundant to investigate SNPs not found in blocks, but since our final goal is to compare all methods we assigned the same cut-off for blocking and single SNP in order to remove potential bias. By employing the same significant threshold as single SNP analysis, haploblock analysis loses its inherent advantage of decreased comparisons. A previous study indicates Bonferroni correction with haploblocks is anti-conservative for univariate SNP analysis and therefore its significance threshold (p-value) is higher than it should be in reality [Johnson, et al. 2010]. We rationalize over-estimating the threshold for the gold standard (single SNP) since we want ensure the alternative methods (haplotype, gene-based) are truly better.

5.1.6.3 Gene Level Analysis Gene based testing was investigated with all five methods from the previous power analysis (GWiS, Fisher, Vegas, GeneBlock-Gabriel, GeneBlock-HapBlock). Both GWiS and Vegas were implemented on the Frank computer server. Computation time was dramatically decreased by running at a chromosome level in parallel and by only allowing 1,000,000 total permutations. Fisher and both GeneBlock methods employed an algorithm (Figure 15) which maximized the abilities of Pittgrid to allow 1000 submissions at a given time. A p-value threshold for significance was determined with a Bonferroni correction at 0.05 level based on the total number of genes analyzed. Because correlation exists between p-values of nearby genes, this correction is a conservative estimate. Interesting genes were graphed in LocusZoom to give a plot with p-values [Pruim, et al. 2010] LD and recombination rates visualized within plots is from HapMAP II CEU data [Consortium 2003]. A few non-significant

SNPs failed to map to LocusZoom HapMAP II CEU data and therefore were excluded from the plots. Haploview LD plots were also generated for some plots to give a more complete illustration of LD structure

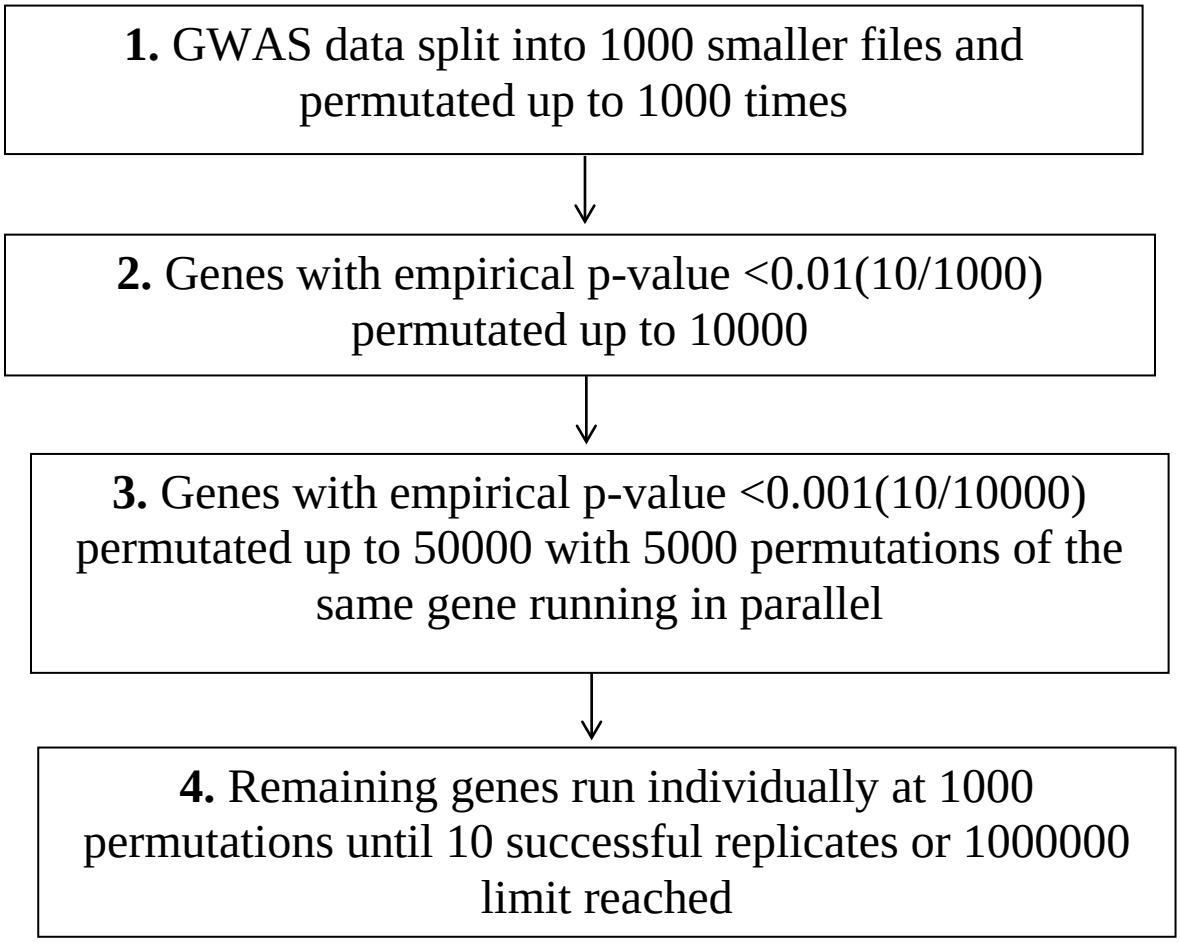

Figure 15. GeneBlock and Fisher Algorithm on PittGrid

5.2 RESULTS

5.2.1 *APOE*

Single-SNP analysis confirmed previous findings of a strong association between *APOE* and AD [Bertram and Tanzi 2009; Kamboh, et al. 2012b]. Twenty-five SNPs in or ±25kb of the *APOE* region were found statistically significant at p< 5e-8. Rs4420638 has the lowest overall p-value at 5.38Ee-58. Since *APOE* is already well established in the literature we excluded it from the subsequent analyses to reduce computation burden. High linkage near *APOE* causes nearby genes to demonstrate an artificial association [Li, et al. 2008; Yu, et al. 2007]. Whether these genes have an independent association remains controversial, but certainly they are strongly driven by *APOE's* association in gene level analysis [Cervantes, et al. 2011; Roses, et al. 2010]. We therefore exclude all genes within 50kb of *APOE (PVRL2, TOMM40, APOC4, APOC2, APOC1).* Figure 16 gives a gene map of ±200kb for *APOE* based on hg18 from the software LocusZoom [Pruim, et al. 2010]. Note, the Plink provided hg18-list does not include the gene *APOC1P1* and it was therefore excluded from the gene level analysis.

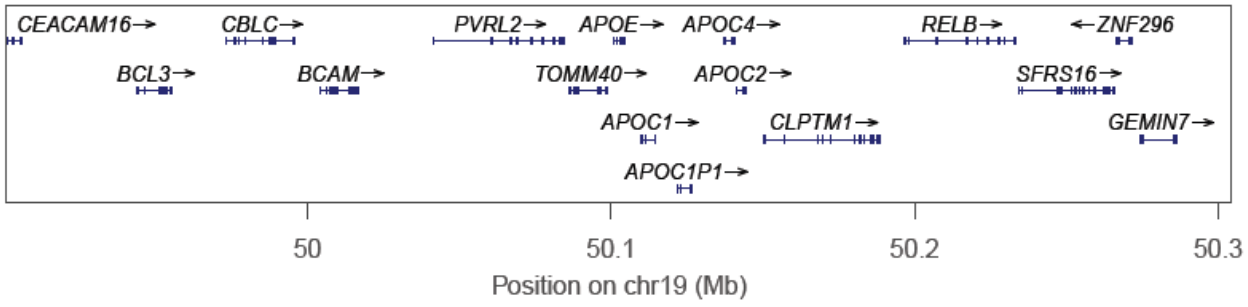

Figure 16. APOE Region (±200kb)

5.2.2 Single-SNP Analysis after exclusion of *APOE* region

In total, 707,247 SNPs were left after exclusion of the *APOE* region. Investigation for determining a Bonferroni cut-off revealed 119,682 blocks and an additional 90,406 SNPs outside block boundaries for a total of 210,088 independent markers. No SNPs achieved a genome-wide significance of 2.3e-7 (0.05/210,088) but a few were highly suggestive. A QQ plot and a Manhattan plot of GWAS results are presented in Figures 17 & 18. Table 6 shows a list of the top ten SNPs from analysis. QQ plot reveals p-values strongly deviating from expect values under a uniform distribution indicating population stratification is likely.

Table 6. Ten SNPs with Lowest P-value

SNP	Chr	BP	Nearest Gene#	P-value	OR	MAF*
rs496990	18	74818363	*SALL3*(-22.9kb)	6.29E-07	1.797	0.06
rs312834	17	72894656	*SEPT9*	6.91E-07	0.7552	0.34
rs10854698	22	35910084	*SSTR3*(-22kb) *C1QTNF6*	1.30E-06	0.771	0.45
rs12616462	2	212198517	*ERBB4*	1.31E-06	1.484	0.38
rs2456955	7	113025148	-	1.34E-06	1.45	0.14
rs1011158	8	9142241	-	1.98E-06	1.297	0.49
rs93075	17	72882063	*SEPT9*	2.14E-06	0.7639	0.34
rs2520297	7	113041997	-	2.79E-06	1.442	0.14
rs214380	20	53525617	-	3.62E-06	1.287	0.38
rs17213073	6	9445254	-	3.74E-06	1.34	0.23

*MAF taken from non-imputed data set
#Genes annotated with Plink glist-h18

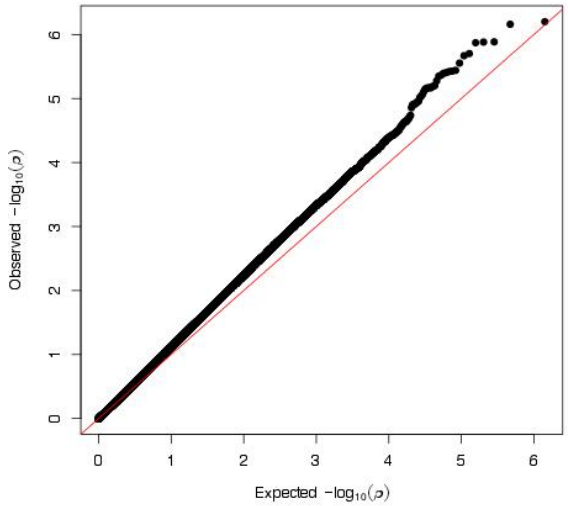

Figure 17. QQ Plot of SNPs in Alzheimer GWAS

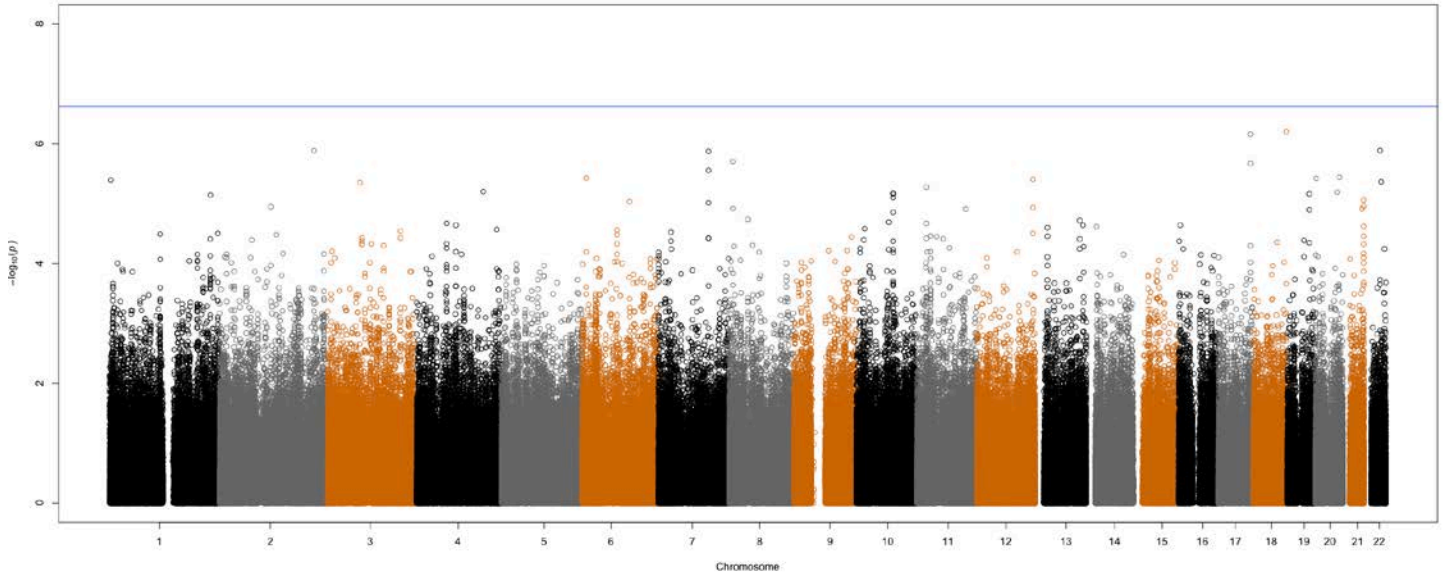

Figure 18. Manhattan Plot of Alzheimer GWAS

Rs496990 located upstream of *SALL3* (22.9kb) showed the strongest association (p= 6.29E-07). *SALL3* is a strongly conserved zinc finger protein involved in embryonic development [Kohlhase, et al. 1999; Sweetman and Munsterberg 2006]. Though no direct link between *SALL3* and AD currently exists, it is involved in regulation of neurofilament expression levels, indicating a strong potential role for pathogenesis [Baba, et al. 2011]. Two of the top ten SNPs are located in *SEPT9. SEPT9* belongs to the septin family of proteins, which are cytoskeleton proteins involved in many cellular processes [Estey, et al. 2011]. *SEPT*9 is of particular interest because mutations within this gene cause hereditary neuralgic amyotrophic (HNA) [Kuhlenbaumer, et al. 2005]. HNA is an autosomal dominant neurological disorder characterized by weakness and sensory loss in the arm muscle, indicating a potential functional role of *SEPT*9 within the brain [Kuhlenbaumer, et al. 2005]. Both *ERBB4* and *SSTR3* have directly been linked with AD. *ERBB4* is a key regulator of Neuregulin-1 which is involved with synaptic plasticity in the brain [Woo, et al. 2010]. Furthermore, *ERBB4* shows higher expression in neurons for Alzheimers cases compared to age-matched controls [Woo, et al. 2011]. *SSTR3* encodes a receptor for the neurotransmitter somatostatin and is overexpressed in the cortex of Alzheimer patients [Kumar 2005]. *C1QTNF6* is less characterized than the other genes and its function remains unknown, though it may play a role in tumor angiogenesis [Takeuchi, et al. 2011].

5.2.3 Haplotype Analysis

Haploblock analysis revealed 119,682 blocks spread across all autosomal chromosomes. Independent SNPs not found in blocks were not included in this analysis since they were already discussed above. We used the same 2.38e-7 threshold in block analysis as in the single-SNP

GWAS analysis. Three blocks contained statistical significant genes with a p-value below 2.38e-7 (Table 7).

Table 7. Ten HaploBlocks with Lowest P-value

NSNP	NHAP	CHR	BP1	BP2	SNP1	SNP2	P	Gene
5	3	17	21375153	21437192	rs11654214	rs6587170	**9.64E-16**	*C17orf51*
9	10	11	49313619	49414631	rs3862342	rs7113075	**1.25E-13**	*LOC729960**
2	3	4	3211958	3212070	rs362305	rs362304	**6.06E-08**	*HTT* *C4orf44*
3	3	4	69718710	69722553	rs844342	rs861340	2.73E-07	*UGT2B10*
2	3	3	12877009	12877199	rs1508758	rs1848466	2.90E-07	*RPL32*
8	6	12	90597385	90685105	rs6538290	rs10859200	4.15E-07	*MDN1* *CASP8AP2* *GJA10*
10	4	4	85386430	85414739	rs28394162	rs17008469	7.83E-07	-
108	17	6	32772436	32782203	rs2647012	rs35332745	7.89E-07	-
2	3	20	59893658	59899889	rs4925308	rs6061883	1.02E-06	*CDH4*
14	5	2	238091933	238111907	rs3751109	rs13425580	2.08E-06	*LOC93349*

*pseudogene (no coding ability)

A 5 SNP block located in the *C17orf51* contains three predicted haplotypes which gives a p-value=9.46E-16. C17orf51 is an uncharacterized protein at chromosome 17p11.2. Little is known of its effect, but the region has been implicated in linkage analysis with Charcot-Marie-Tooth neuropathy type 1a, indicating a potential mechanism for increased AD association [Raeymaekers, et al. 1991]. After further investigation of haplotypes within this block it is becomes clear that a rare haplotype 22122(2-minor allele) is largely driving the association. Haplotype 22122 has an estimated frequency of 0.045 in controls and 0.005 in cases indicating a protective role.

A 9 SNP block (p-value=1.25E-13) located on Chr 11 had no known gene in glist-hg18, but further review found it located in NADPH oxidase 4 pseudogene (LOC279960). Pseudogenes are non-functional therefore the exact mechanism this haplotype remains elusive.

Once again a rare haplotype (2122222222) is driving this p-value with a frequency of 0.008 in affected and a 0.047 in controls. The final block achieving statistical significance (p-value=6.06E-8) is composed of 2 SNPs and located in the 3'UTR of *HTT* on chromosome 4. *HTT* is the causal gene of Huntington's disease a severe neurodegenerative disorder [HDCRG 1993]. Huntington disease and AD share a similar pathogenesis and therefore maybe involved in similar biological pathways [Dagmar, et al. 2011]. As with the other two significant blocks, a rare haplotype (21) is found to drive the association with 2.9% of the controls containing 21 and only 0.007% of the cases. While, this two SNP haplotype is also located within 25kb of *C4orf44* both the location of haplotype (3'UTR of *HTT*) and the strong relation between Huntington disease and AD lead us to believe the true effect lies with *HTT*.

5.2.4 Gene Level Analysis

For 17,547 genes, genotype data was available for more than one SNP. Accordingly we used a Bonferonni cut-off of 2.85e-6 (0.05/17,547) for the gene-based analyses. Gene level analysis revealed QQ-plots strongly deviate from expectation regardless of method used (Figure 19). Population stratification is the likely culprit as permutation testing is not adequate by itself to control for this stratification. A major concern in gene level analysis involves the ability to properly compare genes of different sizes without bias. Correlation examined between gene-size and -log10 (p-value) was small with a correlation coefficient between -0.07 (GeneBlock-Both) and -0.03 (GWiS) demonstrating permutation testing was effective at controlling for type I error regardless of gene size.

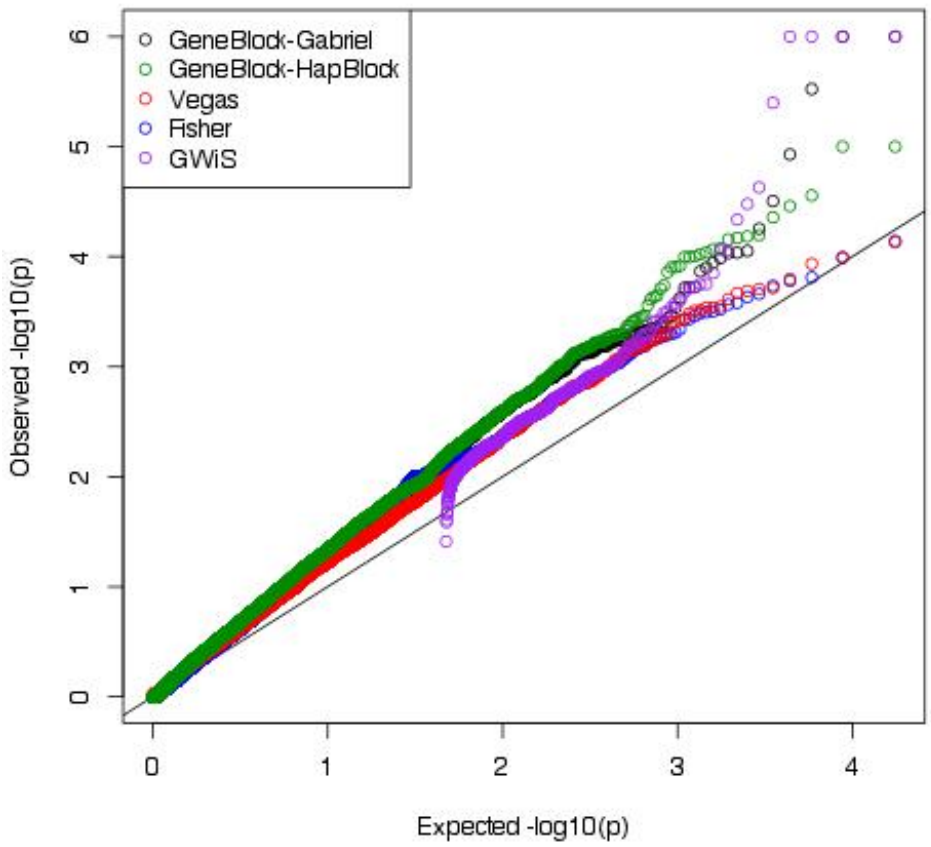

Figure 19. QQ Plot of Gene Base Methods on Alzheimer GWAS

5.2.4.1 Vegas and Fisher As anticipated, Vegas and Fisher produced comparable results. Nine of the top ten genes for each approach were identical when ranking the genes based on lowest p-value, though in slightly different orders (Table 8). However, none of the genes achieved a p-value of statistical significance when accounting for multiple comparisons. The top five loci found in both methods were the same (different order) and included *B4GALT1*, *MAT1A*, *WDR86*, *LOC389435*, and *DNASE1L2*. *B4GALT* encodes an enzyme that catalyzes a reaction to create a N-acetyllactosamine moiety [Ramakrishnan, et al. 2012]. While this specific glycan moiety has not been associated with AD other glycans have [Varki and Freeze 2009]. *MAT1A* is

an enzyme largely involved with methylation through its catalysis of S-adenosylmethionine [Tomasi, et al. 2012]. Mutations within the gene are related to many neurological disorders including AD [Furujo, et al. 2012]. A LocusZoom plot ±25kb with LD and recombination information is available for *B4GALT* and *MAT1A* in Figure 20-21 respectively. Gene plots reveal that *B4GALT1* is largely driven by a few independent SNPs with p-values around 1E-4, while *MAT1A*'s association seems mostly influenced by a LD cluster centered on rs1143694 (p=6.77E-06). No previous link was found between the other three shared genes in the top five with AD and therefore no further analysis was attempted.

Table 8. Top 10 Genes with Fisher and Vegas Methods

Chr	Start-BP	Stop-BP	Gene	SNPs	Vegas P (rank)	Fisher P (rank)
9	33100638	33157356	*B4GALT1*	32	7.40E-05 (1)	0.000102 (2)
10	82021555	82039414	*MAT1A*	38	0.000101 (2)	7.22E-05 (1)
7	150709139	150738057	*WDR86*	36	0.000116 (3)	0.000154 (3)
6	153645074	153645586	*LOC389435*	15	0.00016 (4)	0.000166 (4)
16	2226468	2228713	*DNASE1L2*	16	0.000193 (5)	0.000185 (5)
8	67948883	67976568	*C8orf45*	3	0.000197 (6)	0.000272 (9)
4	88790482	88804534	*DMP1*	14	0.000205 (7)	0.000263 (8)
13	26723691	26728702	*RPL21*	14	0.000214 (8)	0.000369 (15)
14	94727617	94855955	*CLMN*	56	0.000243 (9)	0.000218 (6)
8	27373194	27392730	*CHRNA2*	49	0.000284(10)	0.000302 (10)
16	2213567	2225744	*E4F1*	22	0.000311(11)	0.000232 (7)

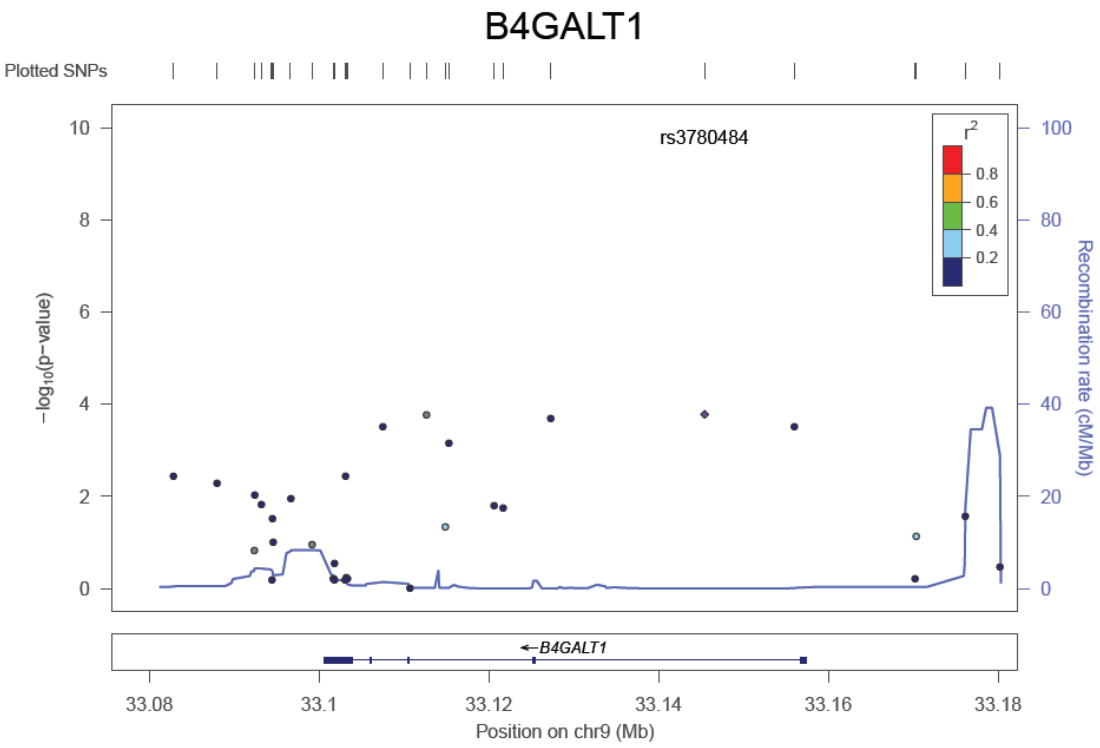

Figure 20. *B4GALT1* P-value Plot

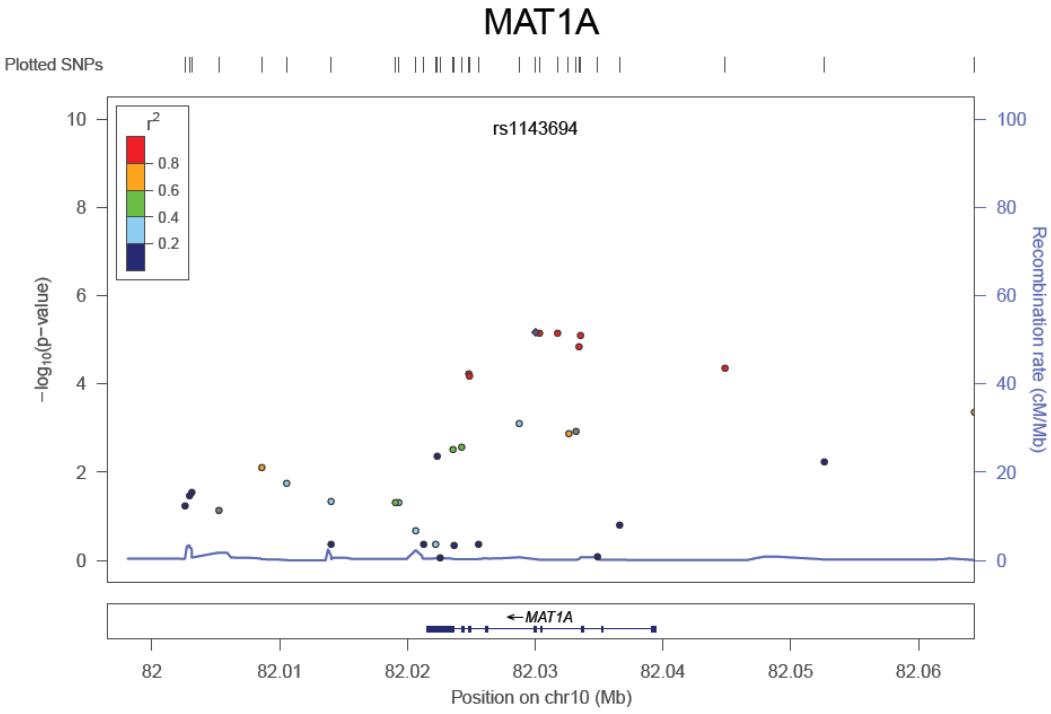

Figure 21. *MAT1A* P-value Plot

5.2.4.2 GWiS GWiS identified four genes with p-values below the suggested Bonferroni cut-off of 2.85e-6 (Table 9). *PRDM16* encodes a zinc finger protein identified as an important regulator for the development of brown adipocytes [Fruhbeck, et al. 2009]. A recent study indicates an additional role in protection of neural cells from oxidative stress, demonstrating a potential mechanism for association with AD [Chuikov, et al. 2010]. *ARHGEF16* is located next to *PRDM16*, but upon further investigation the SNP rs1537404, located in the 3'UTR of *PRDM16* seems to be driving the association (Figure 22). We therefore speculate *ARHGEF16* is not related to AD but happens to be in close proximity to an unrelated strong risk variant. *HLA-DRA* is located in the MHC and is strongly associated with the immune system. Polymorphisms in this gene have already been associated with Parkinson's disease signifying strong potential for a plausible AD relationship [Hamza, et al. 2010; Hill-Burns, et al. 2011]. *TRAF1* encodes a tumor necrosis factor (TRF) receptor, which regulates TNF-alpha [Culpan, et al. 2009]. TNF-alpha is a pro-inflammatory cytokine which plays a pivitol role in many cellular process [Idriss and Naismith 2000]. TNF-alpha and its receptors are both widely implicated for their involvement with AD [Culpan, et al. 2009; Perry, et al. 2001; Swardfager, et al. 2010]. Figur 23-25 reveals gene plots for each significantly associated gene. Contrary to the Fisher and Vegas methods, GWiS is largely driven by a single strongly associated SNP, which is apparent in the gene plots and the large number of the top ten genes overlapping (4) with the six top genes (Table 6) found in the standard GWAs analysis.

Table 9. Top 10 genes with GWiS Method

Chr	Start-BP	Stop-BP	Gene	SNPs	Causal Effects	P-Value
1	2975603	3345045	*PRDM16*	156	2	9.99E-07
1	3361006	3387537	*ARHGEF16*	26	2	9.99E-07*
6	32515624	32520802	*HLA-DRA*	254	4	9.99E-07
9	122704492	122728994	*TRAF1*	105	2	9.99E-07
18	74841262	74859181	*SALL3*	12	1	4.00E-06
17	72789086	73008273	*SEPT9*	81	1	2.35E-05
22	35932190	35938299	*SSTR3*	51	1	3.32E-05
22	35906151	35914276	*C1QTNF6*	46	1	4.61E-05
1	227633615	227636466	*ACTA1*	17	1	8.76E-05
1	227643666	227710711	*NUP133*	25	1	8.76E-05

*Likely driven by proximity to *PRDM16*

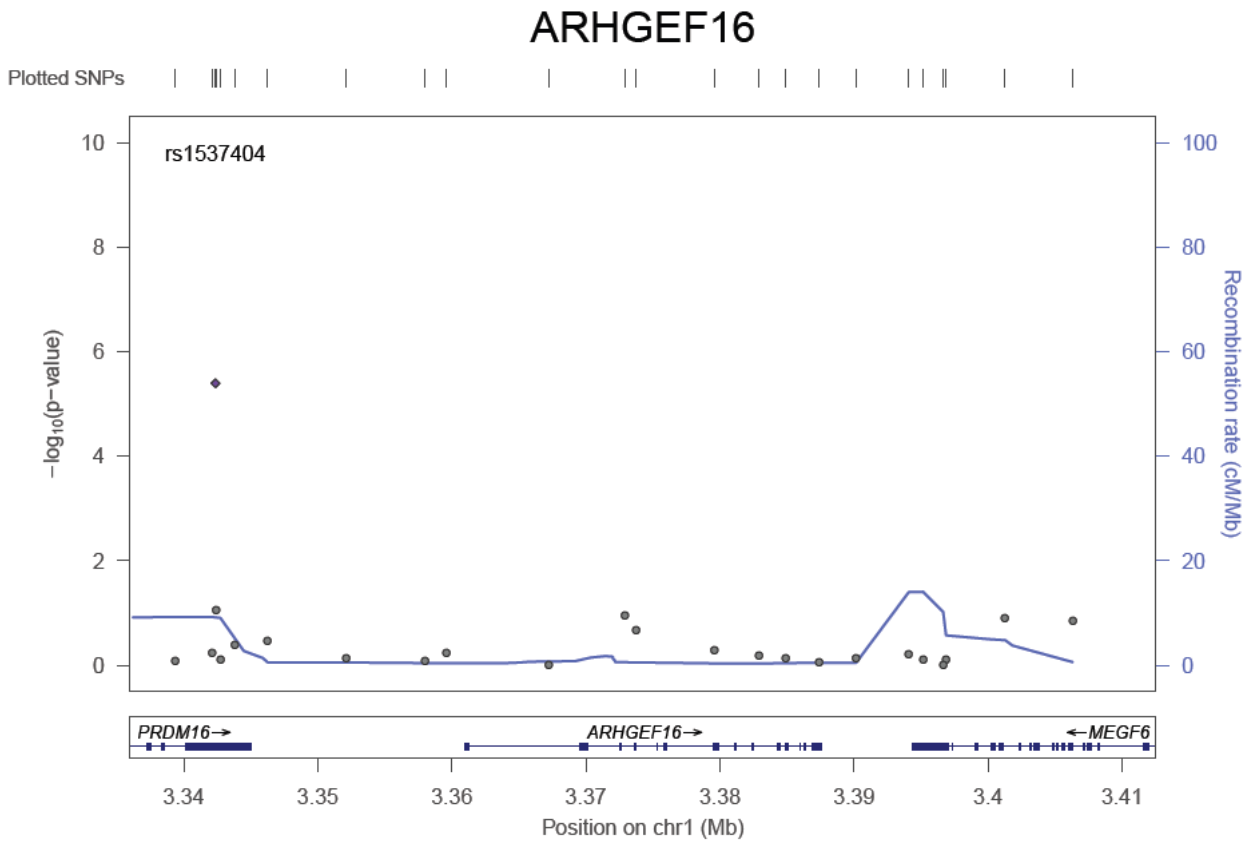

Figure 22. *ARHGEF16* P-value Plot

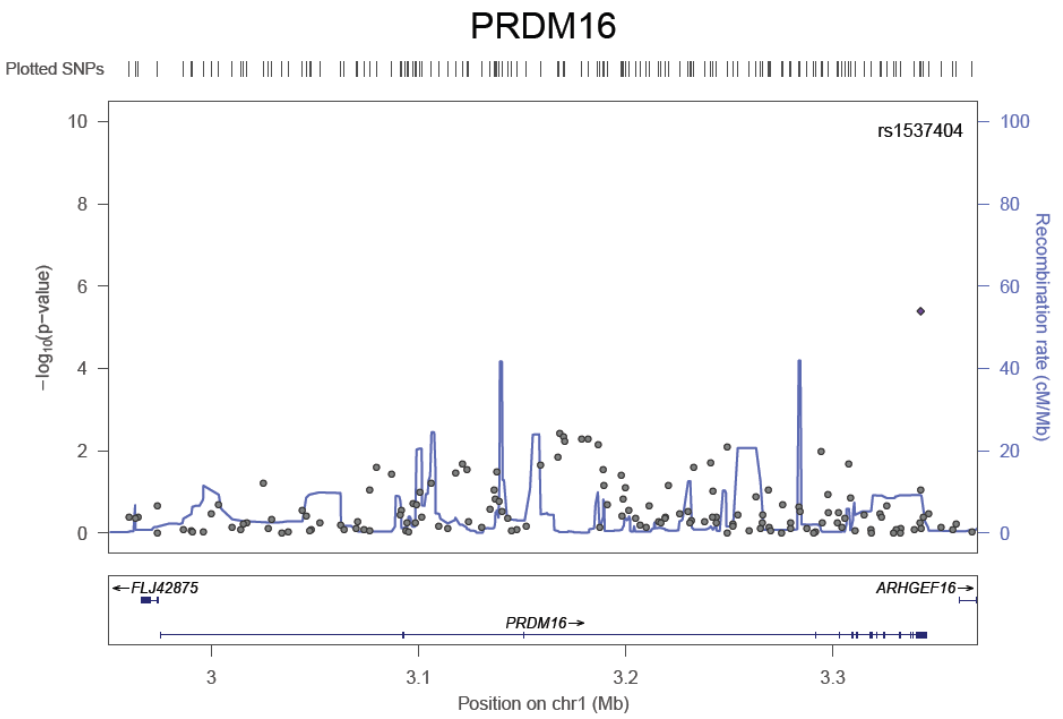

Figure 23. *PRDM16* P-value Plot

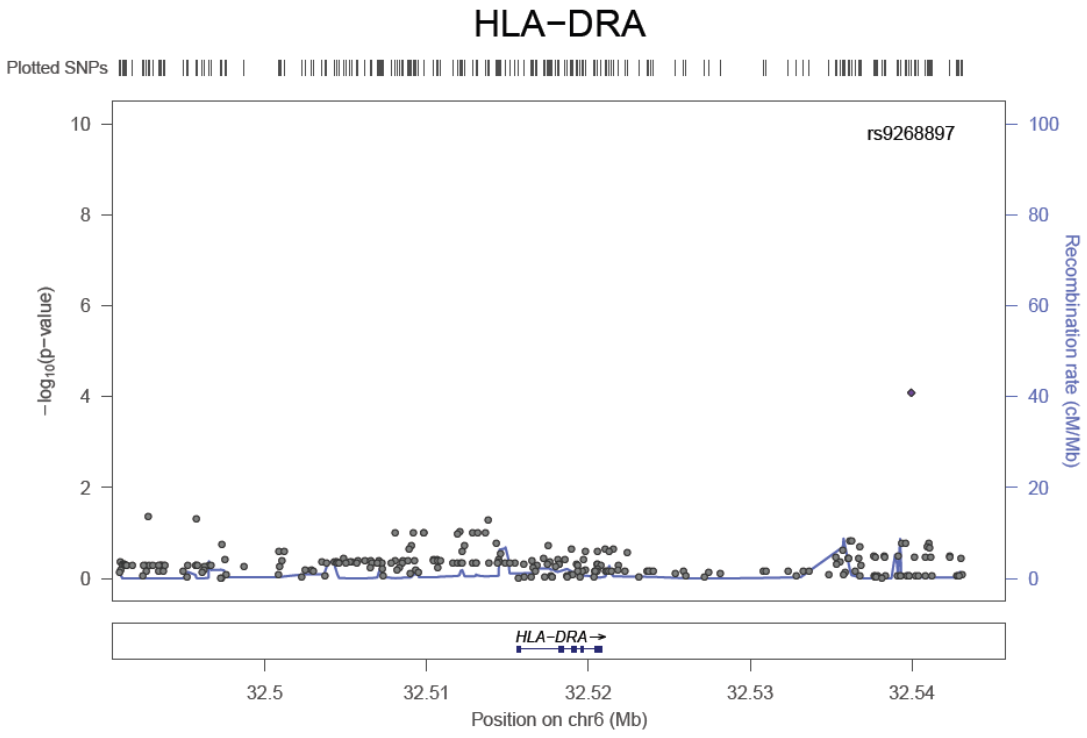

Figure 24. *HLA-DRA* P-value Plot

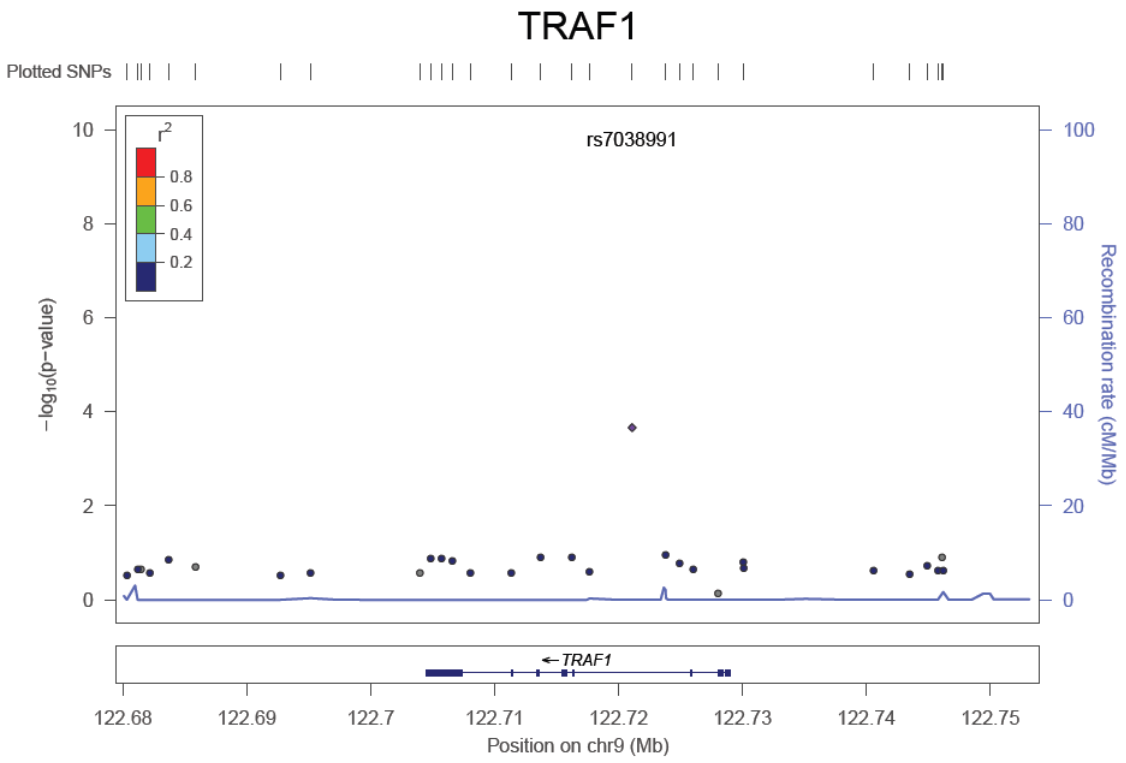

Figure 25. *TRAF1* P-value Plot

5.2.4.3 GeneBlock Despite different algorithms for assigning boundaries, the two GeneBlock methods detected the same top three genes. After the top three genes this concordance disappeared and the rest of the ten highest ranked genes are different. GeneBlock-HapBlock failed to assign block (section 3.2.1) in thirty genes, though none were of significance when compared with GeneBlock-Gabriel. See Table 10 for the top five genes from each method. All genes show fairly strong associations in both methods except *NBEAL2* which is the 4987 ranked gene when using the Gabriel algorithm compared to 4th with the HapBlock. Investigation of the *NBEAL2* haploblocks revealed Gabriel containing one eight SNP block covering the whole gene, where HapBlock has two blocks of four SNPs each. When examining a Haploview LD plot of *NBEAL2* (Figure 26-27) the basis for differing block boundaries becomes apparent, since a tight

LD block is found with the D' method, but when looking at r^2 the second block is not distinguishable [Barrett 2009; Barrett, et al. 2005]. Haplotype regression indicates a p-value of 0.163 for the Gabriel block, while deriving p-values of 0.06 and 4.26e-006 with the HapBlock algorithm. *NBEAL2* is a clear example of how different blocking methods can give varying results.

Table 10. Top 5 Genes in GeneBlock Method

Chr	Start-BP	Stop-BP	Gene	SNPs	Gabriel P (rank)	HapBlock P (rank)
17	21376454	21395500	*C17orf51*	9	9.99E-07 (1)	9.99E-07 (1)
5	138751155	138753504	*MGC29506*	10	9.99E-07 (1)	3.0E-06 (2)
5	138730787	138746900	*SLC23A1*	7	3.0E-06 (3)	8.0E-06 (3)
2	73022672	73152473	*SFXN5*	33	1.18E-05 (4)	1.8E-04 (21)
3	95264544	95328320	*NSUN3*	11	3.11E-05 (5)	0.001 (96)
3	46996176	47026197	*NBEAL2*	8	0.19 (4987)	2.78E-05 (4)
19	58988666	59019460	*NLRP12*	27	4.7E-04(33)	3.47E-05 (5)

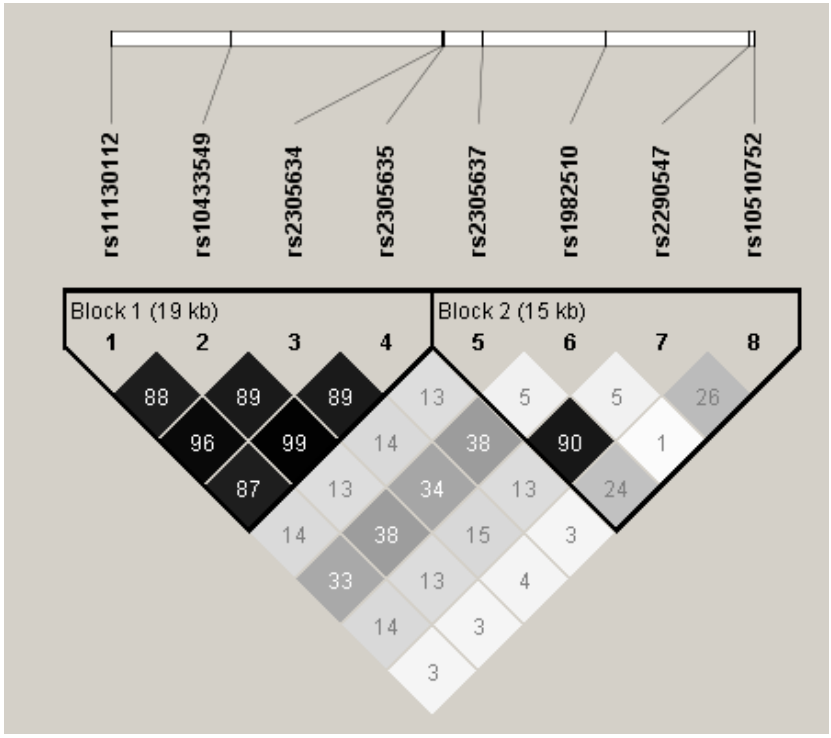

Figure 26. LD (r^2) Plot of *NBEAL2* with HapBlock Boundaries

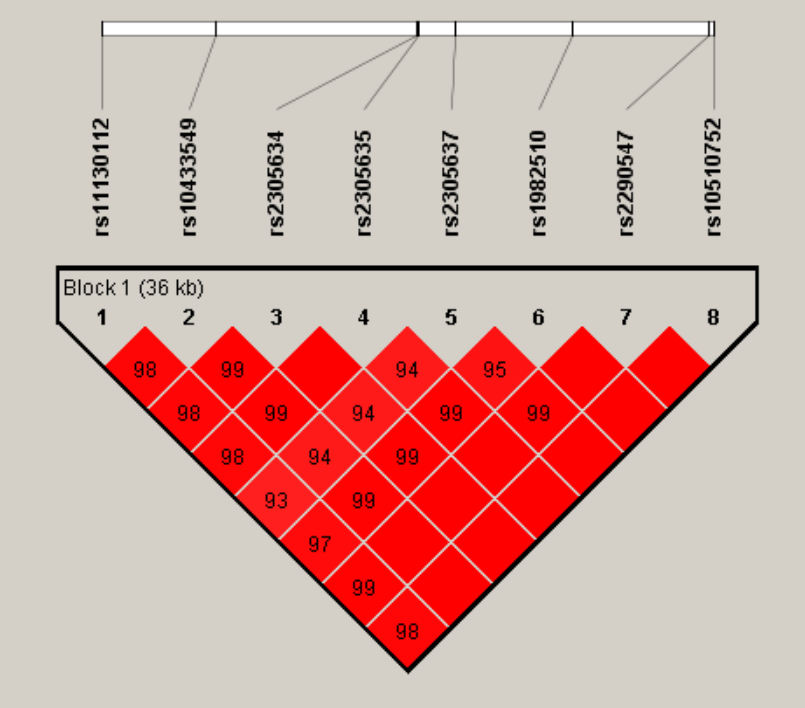

Figure 27. LD (D') Plot of *NBEAL2* with Gabriel Boundaries

It is encouraging that the top three genes were identical for both blocking approaches, but only *C17orf51* was statistically significant in all. *C17orf51* has already been described in section 5.2.3 (HaploBlock Analysis) as an uncharacterized protein with some interesting potential linkage associations. The other two genes (*MGC29506, SLC23A1)* are located next to each other on chromosome 5 and it is difficult to elucidate which gene is driving the signal (Figure 28). *MGC29506* aka *MZB1* codes for a protein which is innate-like B cells involved with calcium storage in the spleen [Flach, et al. 2010]. The effect of this gene in the brain remains elusive. *SLC23A1* encodes for the protein SVCT, which is a key transporter of L-ascorbic acid (vitamin C) throughout the body [Timpson, et al. 2010]. *SLC23A1* is potentially mediating risk through the vitamin C pathway since vitamin C reduces the risk of AD [Morris, et al. 1998; Zandi, et al. 2004]. Further investigation is certainly warranted. Gene plots are available in Figure 28-29.

The gene plots show several genes with no strongly associated SNPs, but when investigating haplotypes these signals become much stronger. Justification therefore exists for GeneBlock method compared to non-blocking analysis. GeneBlock p-values are largely driven by rare haplotype frequencies, similar to results found in the haploblocking method signifying the necessity for replication. Similar with single rare allele analysis, GWAS does not always have adequate power to detect or confirm rare haplotypes associations.

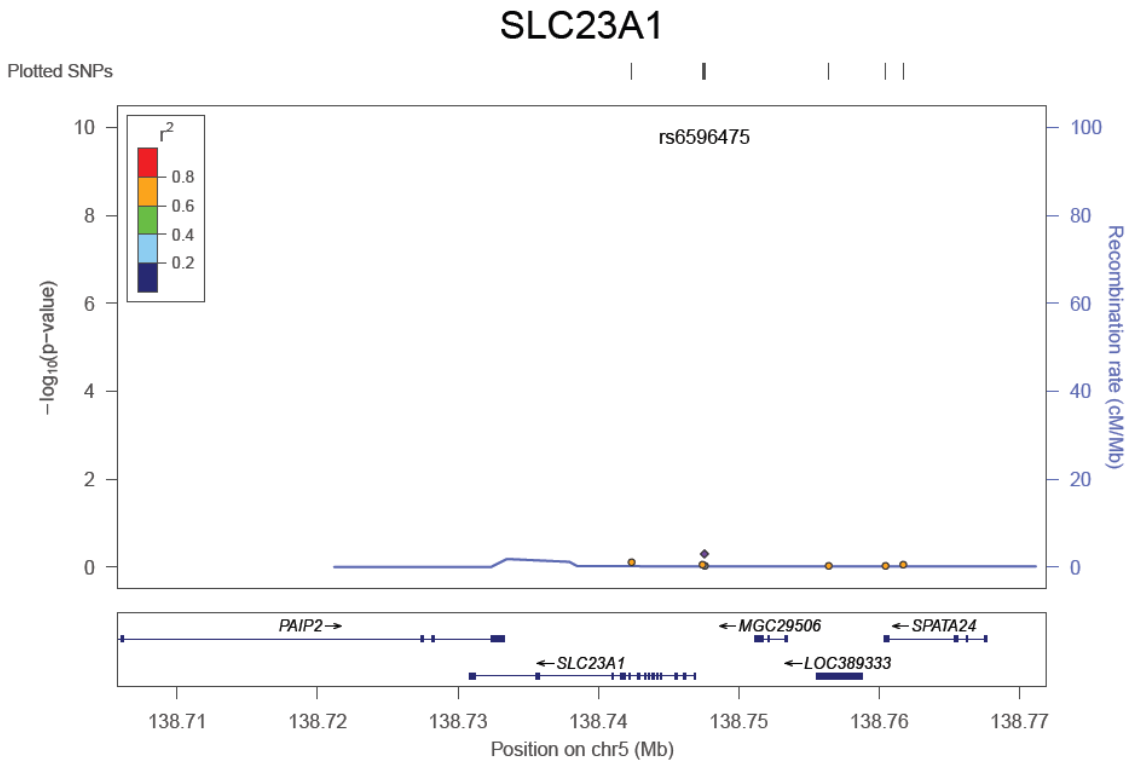

Figure 28. SLC23A1 P-value Plot

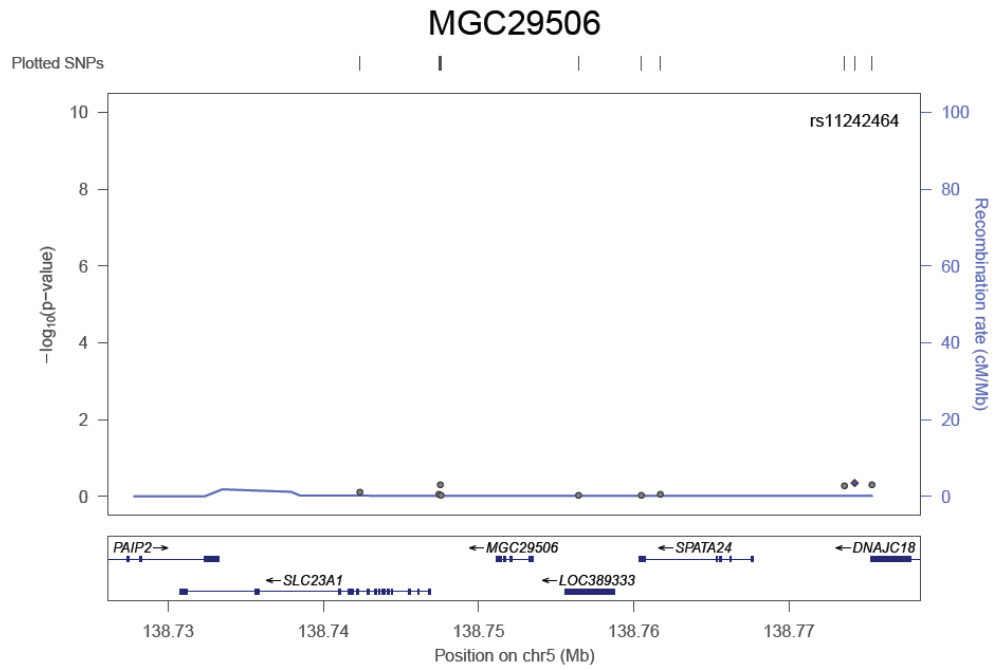

Figure 29. MGC29506 P-value Plot

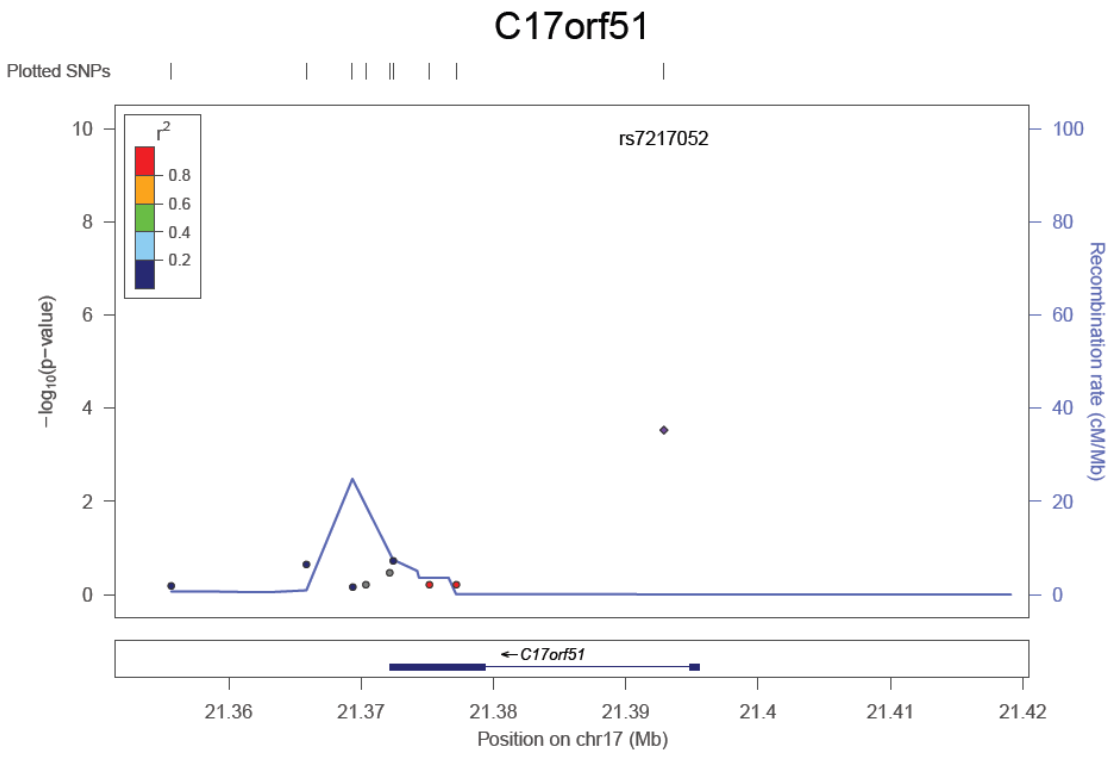

Figure 30. *C1orf51* P-value Plot

5.2.3.4 Comparison of Method Investigation comparing all genes reveals unsurprisingly that Fisher and Vegas seemed highly correlated, while GeneBlock-Gabriel and GeneBlock-HapBlock are also similar. These methods seem to produce almost completely different result when compared with each other. Though Vegas and Fisher method did not find any significant genes, their top genes still seemed interesting since they had many moderate (1E-6<p-value<1E_4) associations relative to either GWiS or GeneBlock largely driven by single associations. This brings up an important point that just because a gene-based test produces a lower p-value, it is not necessarily better at identifying truly intriguing genes.

We performed a rank test to compare how similar the top genes among each method and which genes is best when you combined them. Three dissimilar methods (Vegas, GWiS, GeneBlock-Gabriel) were selected for the analysis. Genes were ranked based on lowest p-value for the three methods and an average was taken. No gene was shared among the top 10 between any of the three methods. This is because each method appears to highlight different associations. Vegas finds genes with many moderately associated p-values, some which are not in high LD. GWiS seems to find genes with solely one strong association. GeneBlock-Gabriel is largely driven by rare haplotype association.

Combing these methods produced interesting results since they each find unique gene structures significant, any gene that is strong in all three methods is worth knowing. See Table 11 for genes with lowest p-value from the three selected gene-based methods. Since GWiS only assigns p-values to 369 genes, all others were equally considered with a rank of 370. Six genes are ranked relatively close with average ranks ranging from 12.33-17.67. Each of these genes were ranked in the top ten of at least one gene-based method. *SEPT9* was the highest ranked gene when combing methods and has previously been described in 5.2.2 as encoding for a

cytoskeleton protein with mutations linked to a neurological disorder. *C8orf45* is an uncharacterized protein or chromosome 8 with an unknown function. *COL18A1* encodes a collagen protein crucial for neural tube closure among other functions [Sertie, et al. 2000]. *RPL21* encodes a ribosomal protein in the 60S subunit of L21. It is highly conserved and considered crucial for protein synthesis within a cell, though no clear mechanism relating with Alzheimer exists [Zhou, et al. 2011]. *USP12* is a histone deubiquitaneses and therefore is involved in a variety of cellular process [Joo, et al. 2011]. *USP12* has never been directly associated with AD, but many other members of the ubiquitine proteasome system have been indicated in neurodegenerative diseases [Dennissen, et al. 2012]. The final gene located in the top cluster belonged to *MAT1A* already described in 5.2.3.1 as being involved in methylation with mutations related to many neurological disorders. Gene plots for these top ranked genes are in Figure 31-34 and Figure 21 (*MAT1A*)

A combination based on p-values was also undertaken by selecting the lowest p-value from Gabriel, Vegas and GWiS (section 4.2.4). The combinational method leads to the most significant genes even when accounting for additional multiple comparisons though the direct Bonferroni correction leads to a threshold of 9.5E -7. Since we only create up to 1,000,000 permutation replicates the minimum p-value produced is 9.99E-7. We therefore assume these are equivalent with an understanding that the Bonferroni correction used in this situation is extremely conservative.

Table 11. Top Ranked Genes when Comparing Methods

Gene	Chr	SNP	Start	End	Gene Rank (P)			
					Vegas	GWiS	GeneBlock Gabriel	Avg
SEPT9	17	81	72789086	73008273	12 (2.9E-4)	6 (2.4E-5)	19 (3.2E-4)	12.3
C8orf45	8	3	67948883	67976568	6 (2.0E-4)	12 (1.8E-4)	22 (3.9E-4)	13.3
COL18A1	21	86	45649524	45758062	18 (3.9E-4)	21 (3.2E-4)	8 (9.1E-5)	15.7
RPL21	13	14	26723691	26728702	8 (2.1E-4)	24 (3.9E-4)	18 (2.7E-4)	16.7
USP12	13	30	26540435	26644029	16 (3.7E-4)	29 (5.9E-4)	6 (5.5E-5)	17
MAT1A	10	38	82021555	82039414	2 (1.0E-4)	11 (1.4E-4)	40 (5.2E-4)	17.7
PROS1	3	8	95074588	95175615	38 (8.6E-4)	43 (1.0E-3)	26 (4.3E-4)	35.7
DMP1	4	14	88790482	88804534	7 (2.1E-4)	56 (1.2E-3)	47 (5.8E-4)	36.7
CRYGN	7	18	150757988	150768032	20 (5.0E-4)	91 (2.0E-3)	39 (5.2E-4)	50
NELL1	11	393	20647711	21553577	110 (2.6E-3)	22 (3.7E-4)	21 (3.7E-4)	51

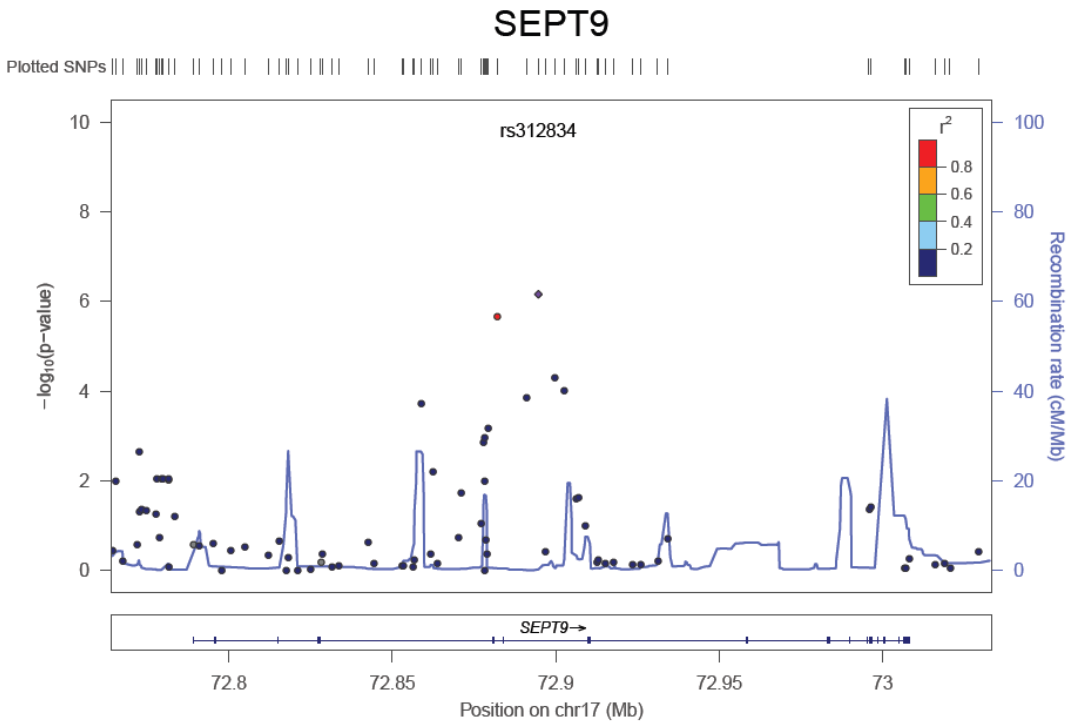

Figure 31. *SEPT9* P-value Plot

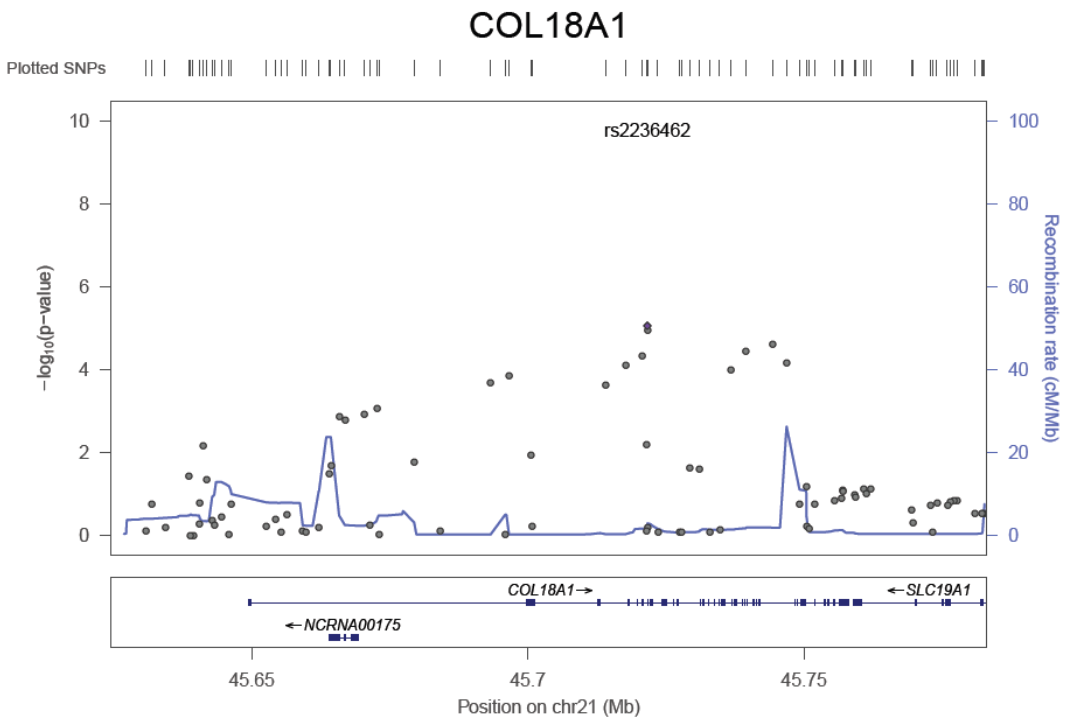

Figure 32. *COL18A1* P-value Plot

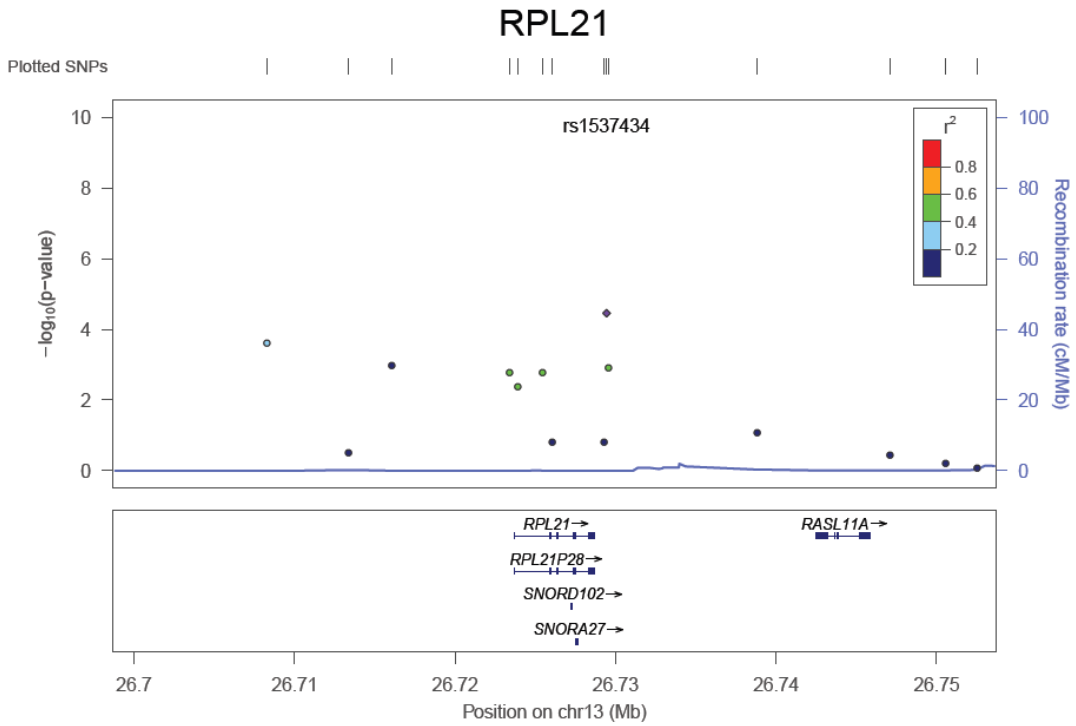

Figure 33. *RPL21* P-value Plot

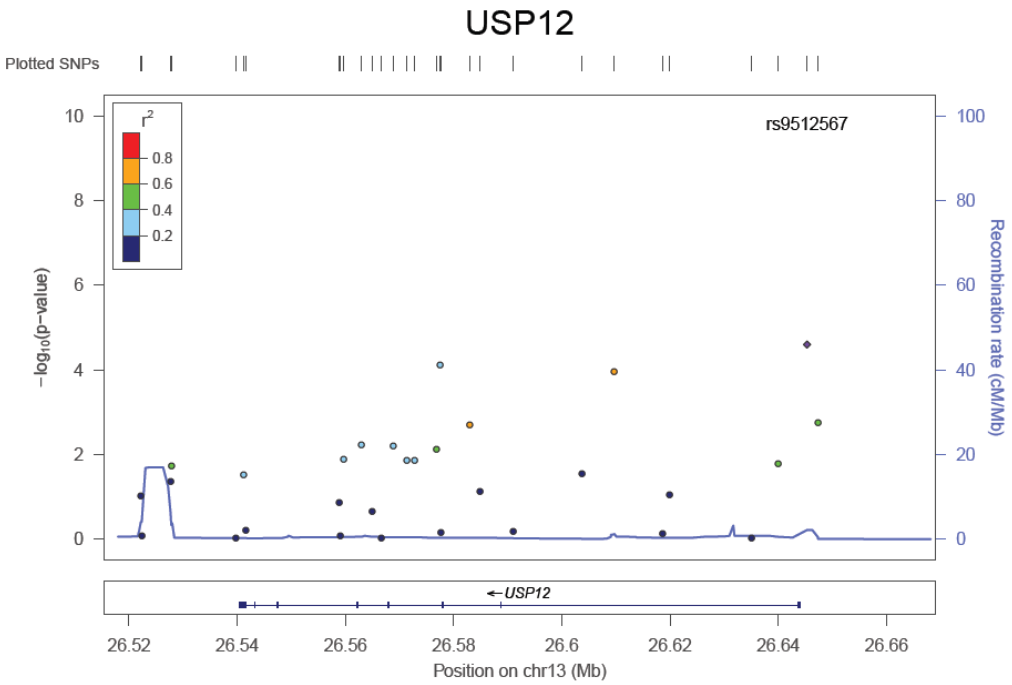

Figure 34. *USP*12 P-value Plot

5.2.5 Comparison of SNP, Haplotype, and Gene Level Analysis

We explored relationships between the different levels of analysis (single-SNP, haplotype, gene) by examining the top 100 ranked genes from each method. Only one gene (*SEPT9*) is shared among the top ten of the single SNP analysis, haploblock, and gene level analysis (combination method). About one-third of the top 100 genes are shared among all three methods. Figure 35 shows a Venn diagram comparing the gene lists from the three methods. Single SNP analysis contained 102 genes since there was a tie for the top 100 genes. Interestingly almost all of the genes shared between two methods are present when comparing all three suggesting these genes are likely by the same risk variant(s). This **analysis** finds that about two-thirds of the top 100 genes are unique to that specific level of analysis indicating the value of investigating each.

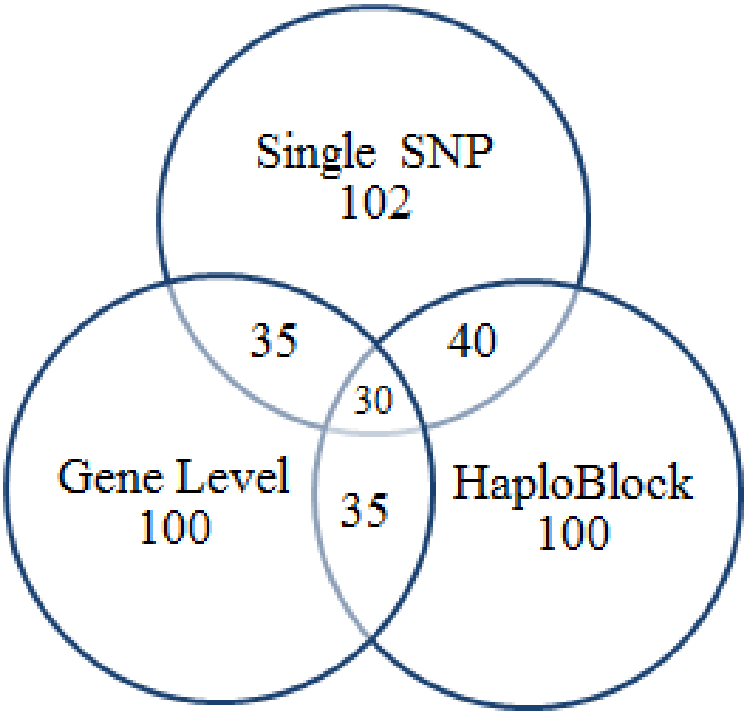

Figure 35. Venn Diagram Comparing Different Levels of GWAS Analysis

5.3 SUMMARY

We investigated an AD GWAS consisting of 1334 cases and 1475 controls. The GWAS was analyzed with a standard single SNP approach, a haplotype (Gabriel) based analysis, and 4 gene-based methods (Vegas, Fisher, GeneBlock,GWiS). The *APOE* region on chromosome 19 was excluded from analysis since it is already well established in previous literature and would take considerable computational resources in the gene-based permutation testing [Coon, et al. 2007; Yu, et al. 2007]. A total of 707,335 markers were included in the final analysis. Single SNP testing found no significant genes. QQ plots of SNPs in the AD GWAS p-values showed likely population stratification, which could not be controlled for with the standard principal

component approach because GWiS does not allow for covariates in its analysis. Haplotype analysis found three block achieving genome-wide significance (p< below 2.38e-7). Two of these are in genes with no predicted relation to AD (*LOC729960*, *C170rf51)*, but a 2 SNP haploblock located in the 3'UTR of *HTT* has potential for a biological meaningful association. Gene-Level analysis found 7 genes which achieved genome-wide significance. Fisher and Vegas identify no significant genes in the AD GWAS, while GWiS and Geneblock identified four (*PRDM16, ARHGEF16, HLA-DRA, TRAF1*) and three (*C17orf51, MGC29506, SLC23A1*) respectively. A combination of the lowest p-values for each gene from Vegas, GeneBlock-Gabriel, and GWiS identified all seven genes at a genome-wide significant level even when considering additional multiple comparisons. *PRDM16, HLA-DRA, TRAF1*, and *SLC23A1* are all involved in pathways associated with AD. Examination of the top 100 genes between the different levels of analysis (single-SNP, haplotype, gene) found about 1/3 are shared, likely driven by the same risk variants

6.0 DISCUSSION AND CONCLUSION

6.1 DISSERTATION CONCLUSION

Investigation of GeneBlock found it performed well compared with other methods in power analysis though it was slightly less powerful than GWiS. When GeneBlock was applied to the AD GWAS it detected several unique genes (*C17orf51, MGC29506, SLC23A1*), achieved genome-wide significance. A combinational approach involving GeneBlock and two other methods (Vegas, GWiS) found four more genes (*PRDM16, ARHGEF16, HLA-DRA, TRAF1*), which achieved Bonferroni corrected significance even when accounting for additional testing. Comparisons between gene-based testing, single SNP, and haplotype analysis found about two-thirds of the top 100 genes from each level of analysis map uniquely, indicating additional value in running not only gene-level, but haplotype level analysis in GWAS. We therefore recommend the use of GeneBlock for gene level analysis by itself or in conjunction with other methods.

6.2 LIMITATIONS

The power analysis presented in this paper was well planned, but in my opinion, completely inapplicable to real GWAS data. Too many potential genetic models exist to truly get an accurate assessment of each models true power. We make estimates about gene size, risk effect, number of risk loci, and correlation among disease SNPs, any of which could drastically alter results if changed. For example, we assume three independent disease causing SNPs per gene because of

the Huang et, al study predicts that many in disease genes, but certainly not all disease genes have three independent risk loci [Huang, et al. 2011]. Our study is not designed to handle deviations from these set parameters, as it is extremely time consuming to just look at the given situation. This is not specific to our power analysis, but a general statement about all gene-based power analyses. I therefore take the results from the power analysis very lightly and understand that the results only apply to a very select set subset of true genetic models. In the AD GWAS, we include 495 controls with significantly lower age and a smaller percentage of females. We justify this by the additional power they bring to the analysis, but because they are not matched like the other controls they could also increase the chances of confounding.

6.3 FUTURE WORK

GeneBlock is a novel method with many potential avenues for optimization. In order to improve the power of the GeneBlock test statistic, I suggest investigating a truncated approach for the Fisher method, which shows higher power when a small number of the overall p-values are significant within the test [Neuhauser and Bretz 2005; Zaykin, et al. 2002]. Both a top n% or using a significant p-value threshold have been suggested for gene-based testing, but neither has been directly compared allowing for multiple potential avenues for truncation [Liu, et al. 2010; Purcell, et al. 2007b]. In theory, implementation of a truncation approach is simple since all of the above GeneBlock code can be reused with a simple filter step.

Currently, GeneBlock is computationally slow, but it could be modified to work on the sample multivariate normal distribution simulation as proposed by Vegas [Liu, et al. 2010]. The largest obstacle is determining correlation between blocks. Certainly a method could be derived

to directly determine relation between all SNPs within each block, but a more efficient procedure involves the use of selecting the top tagging SNPs within each block [Zhang, et al. 2005]. TagSNPs already have defined measures of LD, and correlation between block can be assessed by investigation LD between each tagSNP. GeneBlock could be easily modified to work directly in Vegas for this situation by assigning a given haploblock p-value to its corresponding tagged SNP.

GeneBlock may have additional advantages in analyzing imputed GWAS data. Imputation allows for the estimation of ungenotyped SNPs in a study population from a genotyped haplotypes within a reference population [Howie, et al. 2012]. Imputed datasets can increase the number of SNPs from hundreds of thousands to millions. Undertaking gene-based testing in imputated data is difficult because it is heavily influenced by the haplotype reference population haplotypes, and therefore LD measures within a gene will be biased. Investigating a haploblock level as proposed in GeneBlock, could eliminate much of this bias since a single p-value is produced within each haploblock. A simulation would be derived to test this hypothesis.

APPENDIX A

TABLES AND FIGURES

Table A1. Large Sample Power Analysis Results

TEST	Gene	Type I Error	Power
Gabriel	***SYPL2***	**0.012**	**0.99**
Hapblock	*SYPL2*	0.005	0.968
Fisher	*SYPL2*	0.01	0.426
GWIS	*SYPL2*	0.07	0.989
VEGAS	*SYPL2*	0.09	0.445
Gabriel	*C1orf92*	0.013	0.933
Hapblock	*C1orf92*	0.008	0.89
Fisher	*C1orf92*	0.016	0.607
GWIS	***C1orf92***	**0.01**	**0.957**
VEGAS	*C1orf92*	0.013	0.655
Gabriel	*FOXE3*	0.007	0.97
Hapblock	***FOXE3***	**0.009**	**0.976**
Fisher	*FOXE3*	0.012	0.865
GWIS	*FOXE3*	0.01	0.93
VEGAS	*FOXE3*	0.012	0.865
Gabriel	*CACHD1*	0.011	0.929
Hapblock	*CACHD1*	0.009	0.952
Fisher	*CACHD1*	0.014	0.901
GWIS	***CACHD1***	**0.015**	**0.989**
VEGAS	*CACHD1*	0.011	0.9
Gabriel	***TMEM51***	**0.07**	**0.922**
Hapblock	*TMEM51*	0.013	0.86
Fisher	*TMEM51*	0.014	0.826
GWIS	*TMEM51*	0.011	0.906
VEGAS	*TMEM51*	0.016	0.856
Gabriel	*HS2ST1*	0.01	0.959
Hapblock	***HS2ST1***	**0.009**	**0.994**
Fisher	*HS2ST1*	0.011	0.977
GWIS	*HS2ST1*	0.009	0.985
VEGAS	*HS2ST1*	0.012	0.979

Table A2. Higher Replicate Power Assessment Results

		Method					
Gene		VEGAS	GWIS	Fisher	Gabriel	Hapblock	Top Method
SESN2	1	1.00E-04	5.00E-06	5E-05	1.81371E-05	8.523E-05	GWIS
	2	0.00234	4.93E-05	6.17E-05	8.61104E-05	0.0008576	GWIS
	3	0.000386	1.00E-06	1.49E-05	0.000007	0.000008	GWIS
	4	0.000873	0.000388	0.000472	7.99399E-05	0.0007241	Gabriel
	5	0.000705	3.00E-06	1.95E-05	0.000009	2.616E-05	GWIS
	6	0.00206	4.17E-05	1.14E-05	2.34166E-05	0.0007096	Fisher
	7	0.000142	0.000384	6.63E-05	6.19909E-05	1.885E-05	Hapblock
	8	0.000215	1.36E-05	0.000151	9.0689E-05	0.0002394	GWIS
	9	4.50E-05	2.00E-06	1.81E-05	7.1425E-05	2.2E-05	GWIS
	10	0.00109	2.70E-05	3.32E-05	0.000131968	9.882E-05	GWIS
GJA4	1	6.00E-06	9.82E-05	5.57E-05	1.72286E-05	0.000007	Hapblock
	2	1.20E-05	1.08E-05	7.47E-05	0.000002	0.000001	Hapblock
	3	0.00519	0.001417	2.35E-05	0.000282574	0.0001576	Fisher
	4	1.90E-05	9.00E-06	0.000275	0.000003	9.99e-07	Hapblock
	5	0.000103	2.46E-05	1.75E-05	0.000169239	2.726E-05	Fisher
	6	0.000159	8.00E-06	1.17E-05	0.000004	0.0004266	Gabriel
	7	5.50E-05	0.000111	9.09E-05	0.000004	0.000003	Hapblock
	8	7.00E-06	2.97E-05	0.000209	0.000607128	0.000001	Hapblock
	9	0.000267	9.47E-05	2.31E-05	5.4484E-05	0.0001511	Fisher
	10	0.000654	4.37E-05	2.02E-05	7.83165E-05	0.0002389	Fisher
LAX1	1	0.001418	0.000423	0.000759	0.000244469	0.0008895	Gabriel
	2	0.00967	0.000132	0.017606	9.18628E-05	0.0006617	Gabriel
	3	0.00704	0.001601	0.008985	0.002025522	0.0043029	GWIS
	4	0.001035	3.88E-05	0.000958	0.008097166	0.0012118	GWIS
	5	0.000178	0.000193	0.000172	0.000781739	4.464E-05	Hapblock
	6	5.70E-05	4.20E-05	5.78E-05	6.44371E-05	3.274E-05	Hapblock
	7	0.000207	0.000898	0.000226	0.000129564	5.969E-05	Hapblock
	8	0.0111	0.00247	0.012422	0.003544842	0.003296	GWIS
	9	0.00246	0.00013	0.001196	0.002688172	0.0009792	GWIS
	10	0.000733	1.18E-05	0.000372	0.000988631	0.0008646	GWIS
FAM7Y2A	1	4.50E-05	9.99e-07	2.6E-05	0.000005	0.000005	GWIS
	2	7.60E-05	1.00E-06	5.67E-05	3.07115E-05	6.987E-05	GWIS
	3	0.000108	1.00E-06	0.00012	2.40331E-05	0.000003	GWIS
	4	0.000209	1.00E-06	0.000111	7.46007E-05	0.000007	GWIS
	5	9.40E-05	1.00E-06	6.19E-05	1.9854E-05	3.225E-05	GWIS
	6	9.99e-07	9.99e-07	0.000001	0.000002	0.000001	GWIS/VEGAS
	7	1.20E-05	1	1.2E-05	1.86507E-05	5.498E-05	Vegas
	8	0.000175	2.00E-06	0.000129	0.000585995	3.153E-05	GWIS

Table A2 Continued

	9	0.000102	9.99e-07	9.14E-05	0.000007	0.0014368	GWIS
	10	6.90E-05	9.99e-07	3.62E-05	0.000132772	0.000001	GWIS
TFAP2E	1	3.80E-05	2.00E-06	6.84E-05	0.00216544	0.0001326	GWIS
	2	9.10E-05	2.00E-06	4.46E-05	9.99e-07	2.503E-05	Gabriel
	3	2.30E-05	1.00E-06	0.000004	0.000001	3.841E-05	Gabriel/GWIS
	4	0.000361	9.99e-07	0.000297	0.000004	3.652E-05	GWIS
	5	5.30E-05	9.00E-06	1.84E-05	0.00001	5.666E-05	Gabriel
	6	1.50E-05	9.99e-07	1.83E-05	8.55044E-05	0.000006	GWIS
	7	7.70E-05	1.35E-05	0.00014	0.000105545	3.201E-05	GWIS
	8	0.000306	0.000107	0.000169	0.000595628	7.498E-05	Hapblock
	9	2.30E-05	4.77E-05	2.95E-05	0.000001	0.000003	Gabriel
	10	3.20E-05	1.00E-06	2.23E-05	0.000001	0.000004	Gabriel/GWIS
LCE1A	1	0.000122	0.000761	8.08E-05	1.0982E-05	8.988E-05	Gabriel
	2	0.000193	0.000542	0.000237	0.000005	7.007E-05	Gabriel
	3	0.00014	0.000336	1.53E-05	3.554E-05	6.346E-05	Fisher
	4	0.000716	0.000538	9.86E-05	0.000700378	3.746E-05	Hapblock
	5	8.30E-05	0.000545	7.7E-05	3.42543E-05	7.242E-05	Gabriel
	6	8.40E-05	0.000257	5.89E-05	1.56711E-05	0.000003	Hapblock
	7	5.00E-06	1.94E-05	0.00002	9.99e-07	3.198E-05	Gabriel
	8	2.40E-05	0.000175	1.03E-05	0.000001	0.0001457	Gabriel
	9	0.000199	0.000385	0.000471	5.3508E-05	0.000002	Hapblock
	10	9.00E-05	0.000341	0.00015	1.69653E-05	0.000007	Hapblock
HISTH2AA	1	9.00E-06	4.00E-06	3.72E-05	3.36655E-05	0.0006538	GWIS
	2	2.10E-05	0.00011	2.35E-05	0.000003	0.000125	Gabriel
	3	5.40E-05	1.63E-05	5.35E-05	0.000140136	2.605E-05	GWIS
	4	4.00E-06	0.001382	3.5E-05	0.000528849	0.0002	Vegas
	5	2.30E-05	0.000464	4.67E-05	1.21463E-05	0.0001617	Gabriel
	6	1.70E-05	0.000138	1.58E-05	0.000004	6.633E-05	Gabriel
	7	2.60E-05	5.00E-06	7.54E-05	7.82356E-05	0.000001	Hapblock
	8	0.000147	7.00E-06	0.000477	0.000124888	0.0010508	GWIS
	9	9.10E-05	1.00E-06	0.000202	0.000103931	0.0028027	GWIS
	10	6.00E-06	4.06E-05	1.52E-05	0.000004	0.0001089	Gabriel
AMY2A	1	2.30E-05	2.56E-05	2.6E-05	1.41992E-05	2.954E-05	Gabriel
	2	2.80E-05	7.75E-05	2.03E-05	0.000165292	2.59E-05	Fisher
	3	3.00E-06	0.000276	0.000004	0.000005	1.325E-05	Vegas
	4	0.001626	0.000226	0.001396	0.000213968	0.000276	Gabriel
	5	0.000524	6.88E-05	0.00065	2.43759E-05	2.09E-05	Hapblock
	6	0.000162	7.43E-05	0.000335	0.000502614	0.0015506	GWIS
	7	1.00E-06	2.03E-05	0.000002	0.000003	0.000002	Vegas
	8	1.20E-05	0.000411	1.28E-05	1.46924E-05	1.403E-05	Vegas
	9	0.000143	0.003343	6.03E-05	0.000163396	0.0004624	Fisher

Table A2 Continued

	10	7.80E-05	0.000116	0.000235	0.000162462	7.552E-05	Hapblock
IVNS1ABP	1	0.000184	0.000195	0.00039	0.000137186	0.0007445	Gabriel
	2	1.10E-05	0.000306	1.3E-05	0.001121202	7.543E-05	Vegas
	3	9.00E-05	0.000636	9.54E-05	8.57449E-05	0.0006682	Gabriel
	4	3.80E-05	0.000195	2.25E-05	0.003685957	0.0001661	Fisher
	5	5.30E-05	1.51E-05	5.15E-05	0.000004	0.000001	Hapblock
	6	9.10E-05	0.000287	0.000101	0.000409082	3.981E-05	Hapblock
	7	8.00E-05	0.000536	7.97E-05	0.000930665	0.0004065	Fisher
	8	1.60E-05	0.000247	2.14E-05	4.47912E-05	1.558E-05	Hapblock
	9	0.000187	0.000708	0.000404	0.00024567	0.000794	Vegas
	10	1.50E-05	0.000125	1.74E-05	5.50334E-05	1.566E-05	Vegas

APPENDIX B

COMPUTER CODE

###Code for all Gene-Based Methods is shown for both real data and simulation data assuming analysis of one gene. This can be easily modified to work with many necessary genes###

3.2 Code for Gene-Based Methods

3.2.1 GeneBlock-Gabriel

###Condor Code to submit to grid###

```
universe = vanilla
Executable =  Rscript.bat
requirements   = (Arch=="INTEL" || Arch=="X86_64")  && (OpSys == "WINNT51" || OpSys
== "WINNT60" ||OpSys == "WINNT61") && HasR == TRUE
should_transfer_files = YES
when_to_transfer_output = on_exit_or_evict
arguments = gene2.r
transfer_input_files = gabriel.r, gene.ped, gene.map ,plink.exe
log = gabriel.log
error = gabriel.error
output= gabriel.out
Notification = Complete
notify_user =
queue
```

```
##Gabriel.r###
unlink("pop.*")
l<-list.files()
z<-grep(".map",l,value=TRUE)
z1<-sub(".map","",z)

v<-grep("number",l,value=TRUE)
```

###Two R functions taken from MADAM and caTools respectively###

```
fishersum<-function (P)
{
   return(sum(-2 * log(P)))
}
sample.split<-function (Y, SplitRatio = 2/3, group = NULL)
{
   nSamp = length(Y)
   nGroup = length(group)
   if (nGroup > 0 && nGroup != nSamp)
      stop("Error in sample.split: Vectors 'Y' and 'group' have to have the same length")
   BinOne = logical(nSamp)
   SplitRatio = abs(SplitRatio)
   if (SplitRatio >= nSamp)
      stop("Error in sample.split: 'SplitRatio' parameter has to be i [0, 1] range or [1, length(Y)]
range")
   U = unique(Y)
   nU = length(U)
   if (2 * nU > nSamp | nU == 1) {
      n = if (SplitRatio >= 1)
         SplitRatio
      else SplitRatio * nSamp
      rnd = runif(nSamp)
      if (nGroup)
         split(rnd, group) <- lapply(split(rnd, group), mean)
      ord = order(rnd)
      BinOne[ord[1:n]] = TRUE
   }
   else {
      rat = if (SplitRatio >= 1)
         SplitRatio/nSamp
      else SplitRatio
      for (iU in 1:nU) {
         idx = which(Y == U[iU])
         idx = which(Y == U[iU])
         n = round(length(idx) * rat)
         rnd = runif(length(idx))
         if (nGroup) {
            grp = group[idx]
            split(rnd, grp) <- lapply(split(rnd, grp), mean)
         }
         ord = order(rnd)
         BinOne[idx[ord[1:n]]] = TRUE
      }
   }
   if (SplitRatio >= 1) {
      n = sum(BinOne) - SplitRatio
```

```
      if (n > 0)
         BinOne[sample(which(BinOne), n)] = FALSE
      else if (n < 0)
         BinOne[sample(which(!BinOne), -n)] = TRUE
   }
   return(BinOne)
}
###Need to establish base cutoff for real population in which simulation is greater###
l1<-paste("plink --file",z1," --logistic --out logistic")
system(l1)
r1<-read.table("logistic.assoc.logistic",header=TRUE)
s<-dim(r1)[1]
l2<-paste("plink --file",z1," --blocks --out data1")
system(l2)
if (length(readLines("data1.blocks"))>0){
r2<-read.table("data1.blocks",fill=TRUE,row.names=NULL)
r2<-r2[,-1]
r2a<-c(as.matrix(r2))
r3<-r2a[which(r2a!="")]
r3a<-as.matrix(r1[,2])
r3b<-match(r3,r3a)
r3c<-r3a[-r3b]
length(r3c)
write.table(r3a[-r3b],"notinhaplo.txt",row.names = FALSE,col.names = FALSE,quote=FALSE)
l3<-paste("plink --file",z1," --hap data1.blocks --hap-logistic --hap-omnibus")
} else {
write.table(r1[,2],"notinhaplo.txt",row.names = FALSE,col.names = FALSE,quote=FALSE)
r3c<-1
}
if (length(r3c)>0){
l4<-paste("plink --file",z1," --logistic --extract notinhaplo.txt --out notinhaplo.txt")
system(l4)
r4<-read.table("notinhaplo.txt.assoc.logistic",header=TRUE)
if (length(readLines("data1.blocks"))==0)
{
p<-r4[,9]
}
else{
system(l3)
r5<-read.table("plink.assoc.hap.logistic",header=TRUE)
p<-c(r4[,9],r5[,9])}
}
if (length(r3c)==0){
system(l3)
r5<-read.table("plink.assoc.hap.logistic",header=TRUE)
p<-c(r5[,9])
```

```
}

##Take fishersum of the real population###
if (length(na.omit(p))==0){
pval<-noquote(cbind(z1,s,"NA","NA",1))
write.table(pval,paste(z1,".genepval",sep=""),row.names = FALSE,col.names =
FALSE,quote=FALSE,append=TRUE)
stop()
}else{
g1<-fishersum(na.omit(p))
}
###Create ped file for simulation###
l5<-paste("plink --file",z1,"--recode --out pop")
system(l5)
r<-read.table(paste("count",z1,".txt",sep=""))
n<-read.table(v)
j=n+999
p<-read.table("pop.ped",colClasses="character")
###run actual simulation###
while (n<=j & r<=9)
 {
###Randomly assign disease status to population###
p1<-sample.split(p[,1],SplitRatio = .5)
p1[which(p1=="TRUE")]<-"1"
p1[which(p1=="FALSE")]<-"2"
p3<-as.numeric(p1)
p2<-cbind(p[,1:5],p3,p[,7:dim(p)[2]])
write.table(p2,"pop.ped",row.names = FALSE,col.names = FALSE,quote=FALSE)
###Run new population###
l3<-paste("plink --file pop --hap data1.blocks --hap-logistic --hap-omnibus")
if (length(r3c)>0){
l4<-paste("plink --file pop --logistic --extract notinhaplo.txt --out notinhaplo.txt")
system(l4)
r4<-read.table("notinhaplo.txt.assoc.logistic",header=TRUE)
if (length(readLines("data1.blocks"))>0) {
system(l3)
r5<-read.table("plink.assoc.hap.logistic",header=TRUE)
d<-c(r4[,9],r5[,9])
}
else
{
d<-r4[,9]
}
g2<-fishersum(na.omit(d))
}
if (length(r3c)==0)
```

```
{
l3<-paste("plink --file pop --hap data1.blocks --hap-logistic --hap-omnibus")
system(l3)
r5<-read.table("plink.assoc.hap.logistic",header=TRUE)
d<-c(r5[,9])
g2<-fishersum(na.omit(d))
}
###Count if higher than real fisher sum###
if(g2>g1)
{
r=r+1
t=n+1
write.table(t,paste(z1,".hits",n,sep=""),row.names = FALSE,col.names =
FALSE,quote=FALSE,append=TRUE)
}
n=n+1
}
pval<-noquote(cbind(z1,s,r,n,r/n))
write.table(pval,paste(z1,".genepval",n,sep=""),row.names = FALSE,col.names =
FALSE,quote=FALSE,append=TRUE)
unlink("pop.*")
unlink("snplist.txt")
unlink("*.map")
```

3.2.1 GeneBlock-HapBlock

####HapBlock Code for blocking###

###Convert files to proper format for hapblock blocking program###

```
#! /bin/bash
chmod a+x plinkINTEL.exe
./plinkINTEL.exe --file gene --recode --maf .05 --hwe .0001
./plinkINTEL.exe --file plink --logistic --hwe .0001 --maf .05
fgrep -v "CHR" plink.assoc.logistic>p.txt
awk '{if ($9<=1) print $2}' p.txt>snplist$i.txt
rm p.txt
rm plink.assoc.logistic
./plinkINTEL.exe --file gene --recode12 --extract snplist$i.txt --hwe .0001 --maf .05
echo $(cat plink.map|wc -l)>b.txt
awk '{print $2 " " $4}' plink.map>b1.txt
cat b.txt b1.txt>snpgene.pos
echo "1000 " $(cat plink.map|wc -l)>b2.txt
cut -d ' ' -f2,7- plink.ped>b3.txt
cat b2.txt b3.txt>snpgene.ped
```

##Submit to hapblock blocking program###

```
universe = vanilla
Executable = /u/hab45/hapblock/hapblock/HapBlock.exe
requirements   = (Arch=="INTEL" || Arch=="X86_64")  && (OpSys == "WINNT51" || OpSys
== "WINNT60" ||OpSys == "WINNT61" ||OpSys == "WINDOWS")
should_transfer_files = YES
when_to_transfer_output = on_exit_or_evict
arguments = hapblock.txt
transfer_input_files = hapblock.txt,snpgene.pos,snpgene.ped
log = random.log
error = random.error
output= random.out
Notification = Complete
notify_user =
queue
```

###HapBlock Input###

```
2809 ###Number of Samples##
1670 ###Maximum Number of Markers###
1670 ###Maximum Length of Blocks ###
3 1670 snpgene.pos ##Block partionting method for tagSNP selection (Ignored)####
3 snpgene.ped block_outgene.dat ###Input data and output data ###
1 .80 .099 ##Diversity Method , Percent of haplotype to be a block, Minor haplotype frequency#
1 .80  ###Method for Tag Selection (ignored)##
2 ###Specific tagSNPs (none)###
2 ##Permutations (no)###
```

###Hapblock.r Gene Test Submit Code###

```
universe = vanilla
Executable =  Rscript.bat
requirements   = (Arch=="INTEL" || Arch=="X86_64")  && (OpSys == "WINNT51" || OpSys
== "WINNT60" ||OpSys == "WINNT61" ||OpSys == "WINDOWS") && HasR == TRUE
should_transfer_files = YES
when_to_transfer_output = on_exit_or_evict
arguments = hapblock.r
```

```
transfer_input_files =
hapblock.r,gene.ped,plink.exe,gene.map,/u/hab45/GWAS/combined.blocksC17orf51,/u/hab45/G
WAS/snpgeneh/number100000.txt
log = gene.log
error = gene.error
output= gene.out
Notification = Complete
notify_user =
queue


###Hapblock.r####

unlink("new*")
unlink("pop.map")
z<-list.files(pattern=".map", full.names=TRUE)
z1<-sub(".map","",z)
z3<-sub("./","",z1)
z2<-paste("hapblock",z3,".txt",sep="")

c<-paste("plink --file",z3," --recode --hwe .0001 --maf .05 --out new")
system(c)

fishersum<-function (P)
{
   return(sum(-2 * log(P)))
}
sample.split<-function (Y, SplitRatio = 2/3, group = NULL)
{
   nSamp = length(Y)
   nGroup = length(group)
   if (nGroup > 0 && nGroup != nSamp)
      stop("Error in sample.split: Vectors 'Y' and 'group' have to have the same length")
   BinOne = logical(nSamp)
   SplitRatio = abs(SplitRatio)
   if (SplitRatio >= nSamp)
      stop("Error in sample.split: 'SplitRatio' parameter has to be i [0, 1] range or [1, length(Y)]
range")
   U = unique(Y)
   nU = length(U)
   if (2 * nU > nSamp | nU == 1) {
      n = if (SplitRatio >= 1)
         SplitRatio
      else SplitRatio * nSamp
      rnd = runif(nSamp)
      if (nGroup)
```

```
        split(rnd, group) <- lapply(split(rnd, group), mean)
      ord = order(rnd)
      BinOne[ord[1:n]] = TRUE
   }
   else {
      rat = if (SplitRatio >= 1)
         SplitRatio/nSamp
      else SplitRatio
      for (iU in 1:nU) {
         idx = which(Y == U[iU])
         idx = which(Y == U[iU])
         n = round(length(idx) * rat)
         rnd = runif(length(idx))
         if (nGroup) {
            grp = group[idx]
            split(rnd, grp) <- lapply(split(rnd, grp), mean)
         }
         ord = order(rnd)
         BinOne[idx[ord[1:n]]] = TRUE
      }
   }
   if (SplitRatio >= 1) {
      n = sum(BinOne) - SplitRatio
      if (n > 0)
         BinOne[sample(which(BinOne), n)] = FALSE
      else if (n < 0)
         BinOne[sample(which(!BinOne), -n)] = TRUE
   }
   return(BinOne)
}
###Need to establish base cutoff for real population in which simulation is greater###
l1<-paste("plink --file new --logistic --hwe .0001 --maf .05 --out logistic")
system(l1)
r1<-read.table("logistic.assoc.logistic",header=TRUE)
s<-dim(r1)[1]
###Two R functions taken from MADAM and caTools respectively###
t2<-paste("combined.blocks",z3,sep="")
file.copy(t2,"data1.blocks",overwrite=TRUE)
if (length(readLines("data1.blocks"))>0){
r2<-read.table("data1.blocks",fill=TRUE,row.names=NULL)
r2<-r2[,-1]
r2a<-c(as.matrix(r2))
r3<-r2a[which(r2a!="")]
r3a<-as.matrix(r1[,2])
r3b<-match(r3,r3a)
r3c<-r3a[-r3b]
```

```
length(r3c)
write.table(r3a[-r3b],"notinhaplo.txt",row.names = FALSE,col.names = FALSE,quote=FALSE)
l3<-paste("plink --file new --hap data1.blocks --hap-logistic --hap-omnibus")
} else {
write.table(r1[,2],"notinhaplo.txt",row.names = FALSE,col.names = FALSE,quote=FALSE)
r3c<-1
}
if (length(r3c)>0){
l4<-paste("plink --file new --logistic --extract notinhaplo.txt --hwe .0001 --maf .05 --out
notinhaplo.txt")
system(l4)
r4<-read.table("notinhaplo.txt.assoc.logistic",header=TRUE)
if (length(readLines("data1.blocks"))==0)
{
p<-r4[,9]
}
else{
system(l3)
r5<-read.table("plink.assoc.hap.logistic",header=TRUE)
p<-c(r4[,9],r5[,9])}
}
if (length(r3c)==0){
system(l3)
r5<-read.table("plink.assoc.hap.logistic",header=TRUE)
p<-c(r5[,9])
}

##Take fishersum of the real population###
if (length(na.omit(p))==0){
pval<-noquote(cbind(z1,s,"NA","NA",1))
write.table(pval,paste(z1,".genepval",sep=""),row.names = FALSE,col.names =
FALSE,quote=FALSE,append=TRUE)
stop()
}else{
g1<-fishersum(na.omit(p))
}
###Create ped file for simulation###
l5<-paste("plink --file new --recode --hwe .0001 --maf .05 --out pop")
system(l5)
r=0
n<-0
j=n+9999
p<-read.table("pop.ped",colClasses="character")
###run actual simulation###
while (n<=j & r<=9)
 {
```

```
###Randomly assign disease status to population###
p1<-sample.split(p[,1],SplitRatio = .5)
p1[which(p1=="TRUE")]<-"1"
p1[which(p1=="FALSE")]<-"2"
p3<-as.numeric(p1)
p2<-cbind(p[,1:5],p3,p[,7:dim(p)[2]])
write.table(p2,"pop.ped",row.names = FALSE,col.names = FALSE,quote=FALSE)
###Run new population###
l3<-paste("plink --file pop --hap data1.blocks --hap-logistic --hap-omnibus")
if (length(r3c)>0){
l4<-paste("plink --file pop --logistic --extract notinhaplo.txt --out notinhaplo.txt")
system(l4)
r4<-read.table("notinhaplo.txt.assoc.logistic",header=TRUE)
if (length(readLines("data1.blocks"))>0) {
system(l3)
r5<-read.table("plink.assoc.hap.logistic",header=TRUE)
d<-c(r4[,9],r5[,9])
}
else
{
d<-r4[,9]
}
g2<-fishersum(na.omit(d))
}
if (length(r3c)==0)
{
l3<-paste("plink --file pop --hap data1.blocks --hap-logistic --hap-omnibus")
system(l3)
r5<-read.table("plink.assoc.hap.logistic",header=TRUE)
d<-c(r5[,9])
g2<-fishersum(na.omit(d))
}
###Count if higher than real fisher sum###
if(g2>g1)
{
r=r+1
t=n+1
}
n=n+1
}
pval<-noquote(cbind(z1,s,r,n,r/n))
write.table(pval,paste(z3,".genepval",sep=""),row.names = FALSE,col.names =
FALSE,quote=FALSE,append=TRUE)
unlink("snplist.txt")
unlink("*.map")
```

3.2.2 Fisher

```
###Condor Code to submit to grid###

universe = vanilla
Executable =  Rscript.bat
requirements   = (Arch=="INTEL" || Arch=="X86_64")  && (OpSys == "WINNT51" || OpSys
== "WINNT60" ||OpSys == "WINNT61") && HasR == TRUE
should_transfer_files = YES
when_to_transfer_output = on_exit_or_evict
arguments = fisher.r
transfer_input_files=fisher.r,gene.map,gene.ped,plink.exe
log = fisher.log
error = fisher.error
output= fisher.out
Notification = Complete
notify_user =
queue

###Fisher Code written for R, includes Plink Code###

###Remove any previous simulated genes that could be left over###
unlink("pop.*")
###Pull out proper file names##
l<-list.files()
z<-grep(".map",l,value=TRUE)
z1<-sub(".map","",z)
v<-grep("number",l,value=TRUE)

###Two R functions taken from MADAM and caTools respectively###
fishersum<-function (P)
{
   return(sum(-2 * log(P)))
}
sample.split<-function (Y, SplitRatio = 2/3, group = NULL)
{
   nSamp = length(Y)
   nGroup = length(group)
   if (nGroup > 0 && nGroup != nSamp)
      stop("Error in sample.split: Vectors 'Y' and 'group' have to have the same length")
   BinOne = logical(nSamp)
   SplitRatio = abs(SplitRatio)
   if (SplitRatio >= nSamp)
      stop("Error in sample.split: 'SplitRatio' parameter has to be i [0, 1] range or [1, length(Y)]
range")
   U = unique(Y)
```

```
    nU = length(U)
    if (2 * nU > nSamp | nU == 1) {
        n = if (SplitRatio >= 1)
            SplitRatio
        else SplitRatio * nSamp
        rnd = runif(nSamp)
        if (nGroup)
            split(rnd, group) <- lapply(split(rnd, group), mean)
        ord = order(rnd)
        BinOne[ord[1:n]] = TRUE
    }
    else {
        rat = if (SplitRatio >= 1)
            SplitRatio/nSamp
        else SplitRatio
        for (iU in 1:nU) {
            idx = which(Y == U[iU])
            idx = which(Y == U[iU])
            n = round(length(idx) * rat)
            rnd = runif(length(idx))
            if (nGroup) {
                grp = group[idx]
                split(rnd, grp) <- lapply(split(rnd, grp), mean)
            }
            ord = order(rnd)
            BinOne[idx[ord[1:n]]] = TRUE
        }
    }
    if (SplitRatio >= 1) {
        n = sum(BinOne) - SplitRatio
        if (n > 0)
            BinOne[sample(which(BinOne), n)] = FALSE
        else if (n < 0)
            BinOne[sample(which(!BinOne), -n)] = TRUE
    }
    return(BinOne)
}
###Need to establish base cutoff for real population in which simulation is greater###
l1<-paste("plink --file",z1," --logistic --out logistic")
system(l1)
r1<-read.table("logistic.assoc.logistic",header=TRUE)
s<-dim(r1)[1]
p<-r1[,9]

##Take fishersum of the real population###
if (length(na.omit(p))==0){
```

```
pval<-noquote(cbind(z1,s,"NA","NA",1))
write.table(pval,paste(z1,".genepval",sep=""),row.names = FALSE,col.names =
FALSE,quote=FALSE,append=TRUE)
stop()
}else{
g1<-fishersum(na.omit(p))
}

###Create ped file for simulation###
l5<-paste("plink --file",z1,"--recode --out pop")
system(l5)
r<-read.table(paste("count",z1,".txt",sep=""))
n<-read.table(v)
j=n+999
p<-read.table("pop.ped",colClasses="character")
###run actual simulation###
while (n<=j & r<=9)
 {
###Randomly assign disease status to population###
p1<-sample.split(p[,1],SplitRatio = .5)
p1[which(p1=="TRUE")]<-"1"
p1[which(p1=="FALSE")]<-"2"
p3<-as.numeric(p1)
p2<-cbind(p[,1:5],p3,p[,7:dim(p)[2]])
write.table(p2,"pop.ped",row.names = FALSE,col.names = FALSE,quote=FALSE)
###Run new population###
l6<-paste("plink --file pop --logistic --out pop")
system(l6)
r2<-read.table("pop.assoc.logistic",header=TRUE)
###Two R functions taken from MADAM and caTools respectively###
p4<-r2[,9]
if (length(na.omit(p4))==0){
g2<-1
}else{
g2<-fishersum(na.omit(p4))
}
###Count if higher than real fisher sum###
if(g2>g1)
{
r=r+1
t=n+1
write.table(t,paste(z1,t,".hits",sep=""),row.names = FALSE,col.names =
FALSE,quote=FALSE,append=FALSE)
}
n=n+1
}
```

```
pval<-noquote(cbind(z1,s,r,n,r/n))
write.table(pval,paste(z1,n,".genepval",sep=""),row.names = FALSE,col.names =
FALSE,quote=FALSE,append=TRUE)
unlink("pop.*")
unlink("snplist.txt")
unlink("*.map")
```

3.2.3 Vegas Code

###Frank Submission Code##

```
#!/bin/bash
#PBS -m e
#PBS -M hab45@pitt.edu
#PBS -j oe
#PBS -q shared
#PBS -N vegas21
#PBS -l nodes=1:ppn=1
#PBS -l walltime=23:00:00
#PBS -l vmem=4GB
#PBS -S /bin/bash

###Vegas Code###
cd /home/bdiergaarde/hab45/Vegas
module load r
module load perl
###Random controls are 200 randomly selected controls###
plink --bfile data --make-bed --keep randomcontrols.txt --out controls 21
plink --bfile data --logistic --chr 21 --out alz.p21
awk '{print $2,$NF}' alz.p21.assoc.logistic|tail -n +2>alz.p21
./vegas alz.p21 -custom controls21 -chr 21 -out alz21test
```

3.2.4 GWiS Code

###Frank Submission Code##

```
#!/bin/bash
#PBS -m e
#PBS -M hab45@pitt.edu
#PBS -j oe
#PBS -q shared
#PBS -N gwis9
#PBS -l nodes=1:ppn=1
```

```
#PBS -l walltime=23:00:00
#PBS -l vmem=4GB
#PBS -S /bin/bash


##GWiS Code###

cd /home/bdiergaarde/hab45/GWIS/1.1
module load gsl
cd sample

##Format into GWIS format###
plink --bfile gene --chr 9 --make-bed --out pop9

awk '{print $2,$1,$4,$4}' pop9.bim>./chr9.snp.info

plink --bfile gene  --noweb --recodeA --chr 9 --out chr9

sed '1,1d' chr9.raw|cut -d ' ' -f7->p9.raw
python -c "import sys; print('\n'.join(' '.join(c) for c in zip(*(l.split() for l in sys.stdin.readlines() if
l.strip()))))" < p9.raw > plink9.raw

cat plink9.raw|tr ' ' '\t'>chr9.tped

cd ..

rm ./sample/p9.raw
rm ./sample/plink9.raw
rm ./sample/chr9.raw

###Run GWiS input###
./runsample9.sh
fgrep -w SUMMARY ./result/chr9.result.GWiS.Pval.txt>>results9.txt


##runsample9.sh###

#!/bin/bash
NPERM=100000
export NPERM
./GWiS sample/ chr9 plink ./result/ 2707
###2707 is the randomly selected seed number###
```

4.0 Power Analysis

4.2.1 & 4.3.1 Simulation of data sets for both Power Analysis

```
###plink32 is simply plink v1.07 for 32 bit linux###
##Alzheimer's file binary file is stored under name data###
###See paper for description of gene list (glist.txt)###
#! /bin/bash
chmod a+x plink32.exe
###Plink requires association file to annotate SNPs###
./plink32.exe --bfile data --assoc
./plink32.exe --annotate plink.assoc ranges=glist.txt --border 20
awk '{print $2}' plink.annot|sed '1,1d'>snps.txt
awk '{print $NF}' plink.annot|sed '1,1d'>genelist.txt
sed -e 's/([^()]*)//g' genelist.txt |tr '|' ' '>genel.txt
paste -d " " snps.txt genel.txt>genelist.txt
cut -d ' ' -f2- genelist.txt|sed '1,1d'>g.txt
tr ' ' '\n' <g.txt|sort -u|fgrep -wv .>genes.txt
for l in $(cat genes.txt)
do
echo $l $(fgrep -w $l genelist.txt|awk '{print $1}'|tr '\n' ' ')
done >>genesnp.txt
awk '{print $1 "  " NF-1}' genesnp.txt>genecount.txt
###Rcode ###
data<-read.table("genecount.txt")
###For graph convert all gene with lots of SNPs into something that will show up###
r<-data
r[which(r[,2]>150),2]<-151
jpeg(file="GeneCount.jpeg")
hist(r[,2],main= "SNP Count for Gene in Alzheimer's Data Set ",xlab= "Gene Size ")
dev.off()

###Establish Gene size as <50,50-150,>150###
##Find Median in Each Group###

median(data[which(data[,2]<50),2])
##22##
median(data[which(data[,2]>=50 & data[,2]<=150),2])
##71###
median(data[which(data[,2]>150),2])
##208##

###Gene Count for Simulation Chr 1 in hapgen 2###
###Simulate Chr 1, Risk effect and disease locus makes no difference but is required##
mkdir Simulation
./hapgen2 -m ./haplotype+legend_files_CEU_r24/genetic_map_chr1_CEU_b36.txt -l
./haplotype+legend_files_CEU_r24/chr1.ceu.r24.legend -h
./haplotype+legend_files_CEU_r24/hapmap_r24_b36_fwd.consensus.qc.poly.chr1_ceu.phased -
o ./Simulation/chr.out -dl 554636 0 1.2 2.4 -n 25 25
###Convert to plink ###
```

```
./gtool -G --g ./Simulation/chr.out.cases.tags.gen --s ./Simulation/chr.out.cases.sample --ped
./Simulation/out.ped --map ./Simulation/out.map
./gtool -G --g ./Simulation/chr.out.controls.tags.gen --s ./Simulation/chr.out.controls.sample --
ped ./Simulation/out1.ped --map ./Simulation/out1.map
awk '{print $1 " " $2 " " $3 " " $4 " " 1 " " 2}' ./Simulation/out.ped>./Simulation/outa.ped
cat ./Simulation/out.ped|tr '\t' ' '|cut -d ' ' -f7- >./Simulation/outb.ped
paste ./Simulation/outa.ped ./Simulation/outb.ped>./Simulation/out2.ped
cat ./Simulation/out.map>./Simulation/out4.map
awk  -v var=$i '{print 1 " " "SNP"var " " 0 " " $4}' ./Simulation/out.map>./Simulation/out4.map
awk '{print $1 " " $2 " " $3 " " $4 " " 1 " " 1}' ./Simulation/out1.ped>./Simulation/out3a.ped
cat ./Simulation/out1.ped|tr '\t' ' '|cut -d ' ' -f7- >./Simulation/out3b.ped
paste ./Simulation/out3a.ped ./Simulation/out3b.ped>./Simulation/out3.ped
cat ./Simulation/out2.ped ./Simulation/out3.ped>./Simulation/out4.ped
./plink --file ./Simulation/out4 –make-bed  --out ./Simulation/allchr1 --noweb

###Gene Count##
plink --bfile allchr1 --assoc --noweb
plink --annotate plink.assoc ranges=glist.txt --noweb  --border 25
awk '{print $2}' plink.annot|sed '1,1d'>snps.txt
awk '{print $NF}' plink.annot|sed '1,1d'>genelist.txt
sed -e 's/([^()]*)//g' genelist.txt |tr '|' ' '>genel.txt
paste -d " " snps.txt genel.txt>genelist.txt
cut -d ' ' -f2- genelist.txt|sed '1,1d'>g.txt
tr ' ' '\n' <g.txt|sort -u|fgrep -wv .>genes.txt
for l in $(cat genes.txt)
do
echo $l $(fgrep -w $l genelist.txt|awk '{print $1}'|tr '\n' ' ')
done >>genesnp.txt
awk '{print $1 "   " NF-1}' genesnp.txt>genecount.txt

###Gene Simulation-Only one as example (MARCKSL1)###
fgrep -w MARCKSL1  genelist1.txt|awk '{print $1}'>snplist.txt
fgrep -wf snplist.txt plink.assoc|awk '{print $3}'>positionlist.txt
h=$(fgrep -w $(head -n 1 positionlist.txt) ./CEU.0908.impute.files/CEU.0908.chr1.legend|awk
'{print $2}')
z=$(fgrep -w $(tail -n 1 positionlist.txt) ./CEU.0908.impute.files/CEU.0908.chr1.legend|awk
'{print $2}')
k=`echo $h 100000 | awk '{ print $1-$2}'`
l=`echo $z 100000 | awk '{ print $1+$2}'`
awk -v var=$k -v var1=$l '{if ($2>=var  && $2<=var1) print $0 }'
./CEU.0908.impute.files/CEU.0908.chr1.legend>./CEU.0908.impute.files/MARCKSL1.legend
awk -v var=$k -v var1=$l '{if ($1>=var && $1<=var1) print $0 }'
./CEU.0908.impute.files/genetic_map_chr1_combined_b36.txt>./CEU.0908.impute.files/MARC
KSL1.map
c=$(fgrep -wn $(head -n 1 ./CEU.0908.impute.files/MARCKSL1.legend|awk '{print $1}')
./CEU.0908.impute.files/CEU.0908.chr1.legend|awk -F: '{print $1-1}')
```

```
c1=$(fgrep -wn $(tail -n 1 ./CEU.0908.impute.files/MARCKSL1.legend|awk '{print $1}')
./CEU.0908.impute.files/CEU.0908.chr1.legend|awk -F: '{print $1-1}')

###subtract one because of header not in hap file but in legend file. Check to make fgrep did not
pick up anything weird, numbers used in for loop###
for (( i=$(echo $c); i<=$(echo $c1); i++ ));
do
awk -v var=$i 'NR==var{print $0}'
./CEU.0908.impute.files/CEU.0908.chr1.hap>>./CEU.0908.impute.files/MARCKSL1.hap
done

echo 'ID pos allele0 allele1'>z.txt
cat z.txt ./CEU.0908.impute.files/MARCKSL1.legend>z1.txt
cat z1.txt>./CEU.0908.impute.files/MARCKSL1.legend

####MUST change DL######
c=$(head ./CEU.0908.impute.files/MARCKSL1.legend|awk 'NR==3{print $2}')
z=$(head -n 1 snplist.txt)
z1=$(tail -n 1 snplist.txt)

m=0
n=0
for i in {1..1000}
do
while [ $m -eq $n ]
do
./hapgen2 -m ./CEU.0908.impute.files/MARCKSL1.map -l
./CEU.0908.impute.files/MARCKSL1.legend -h ./CEU.0908.impute.files/MARCKSL1.hap -o
./Simulation/single$i.out  -no_haps_output -dl $(echo $c) 1 1.2 1.4 -n 1000 1000|fgrep
"seed"|awk '{print $5}'>c.txt
cat c.txt>>seed.txt
m=$(cat c.txt|tr -d '.' |tr -d 'e' |tr -d '-')
done
n=$m
./gtool -G --g ./Simulation/single$i.out.cases.gen --s ./Simulation/single$i.out.cases.sample --ped
./Simulation/out.ped --map ./Simulation/out.map
./gtool -G --g ./Simulation/single$i.out.controls.gen --s ./Simulation/single$i.out.controls.sample
--ped ./Simulation/out1.ped --map ./Simulation/out1.map
awk '{print $1 " " $2 " " $3 " " $4 " " 1 " " 2}' ./Simulation/out.ped>./Simulation/outa.ped
cat ./Simulation/out.ped|tr '\t' ' '|cut -d ' ' -f7- >./Simulation/outb.ped
paste ./Simulation/outa.ped ./Simulation/outb.ped>./Simulation/out2.ped
cat ./Simulation/out.map>./Simulation/out4.map
awk  -v var=$i '{print 1 " " $2 " " 0 " " $4}' ./Simulation/out.map>./Simulation/out4.map
awk '{print $1 " " $2 " " $3 " " $4 " " 1 " " 1}' ./Simulation/out1.ped>./Simulation/out3a.ped
cat ./Simulation/out1.ped|tr '\t' ' '|cut -d ' ' -f7- >./Simulation/out3b.ped
paste ./Simulation/out3a.ped ./Simulation/out3b.ped>./Simulation/out3.ped
```

```
cat ./Simulation/out2.ped ./Simulation/out3.ped>./Simulation/out4.ped
x=$(fgrep -w $(fgrep -w $(echo $z) plink.assoc|awk 'NR==1{print $3}')
./Simulation/out4.map|awk '{print $2}')
x1=$(fgrep -w $(fgrep -w $(echo $z1) plink.assoc|awk 'NR==1{print $3}')
./Simulation/out4.map|awk '{print $2}')
./plink --file ./Simulation/out4 --recode --transpose --out ./Simulation/sample$i --noweb --from
$(echo $x) --to $(echo $x1)
rm ./Simulation/*.summary
rm ./Simulation/single*

done

cat ./Simulation/sample*.tped>./Simulation/all.tped
cat ./Simulation/sample1.tfam>./Simulation/all.tfam
plink --tfile ./Simulation/all --assoc --out ./Simulation/plink --noweb
sed '1d;1d' ./Simulation/plink.assoc>./Simulation/p.assoc
rm ./Simulation/plink.assoc
rm ./Simulation/sample*
rm ./Simulation/all.*

awk '{print $2 " " $3}' ./Simulation/p.assoc|sort -u >./Simulation/p2.assoc
rm ./Simulation/p3.assoc

for (( c=1; c<=$(cat ./Simulation/p2.assoc|wc -l); c++ ))
do
j=$(awk -v var=$c 'NR==var{print $1}' ./Simulation/p2.assoc)
w=$(awk -v var=$c 'NR==var{print $2}' ./Simulation/p2.assoc)
echo $w $(awk -v var=$j '{if ($2==var) sum+=$6;if ($2==var) count++} END {print
sum/count}' ./Simulation/p.assoc) >>./Simulation/p3.assoc
done


awk '{if ($2>=.05) print $1}' ./Simulation/p3.assoc>maf05.txt

c=$(head ./CEU.0908.impute.files/MARCKSL1.legend|awk 'NR==3{print $2}')

./hapgen2 -m ./CEU.0908.impute.files/MARCKSL1.map -l
./CEU.0908.impute.files/MARCKSL1.legend -h ./CEU.0908.impute.files/MARCKSL1.hap -o
./Simulation/hapmap.out -no_haps_output -dl $(echo $c)  1 1.2 1.4 -n 10000 10
fgrep -wf maf05.txt ./Simulation/hapmap.out.controls.gen>hapmap.out

./gtool -G --g ./hapmap.out --s ./Simulation/hapmap.out.controls.sample --ped
./Simulation/hapmapout.ped --map ./Simulation/hapmapout.map


###Convert to haploview format###
```

```
awk '{print $2 " " $4}' ./Simulation/hapmapout.map>marker.txt
cat ./Simulation/hapmapout.ped|tr '\t' ' '|cut -d ' ' -f-4 >outc.ped
awk '{print $1 " " $2 " " $3 " " $4 " " 1 " " 1}' outc.ped>outa.ped
cat ./Simulation/hapmapout.ped|tr '\t' ' '|cut -d ' ' -f7- >outb.ped
paste outa.ped outb.ped>out2.ped
java -jar Haploview.jar -pedfile out2.ped -info marker.txt -pairwiseTagging -nogui -out
MARCKSL1

###Actual lists###
y=$(cat MARCKSL1.TESTS|wc -l)
k=`echo $y 10 | awk '{ print $1/$2}'|awk '{print int($1+0.5)}'`
k=3
x=0
while [ $x -lt $k ]
do
echo $((1 + RANDOM%($y)))>>z1.list
x=$(sort -u z1.list|wc -l)
done
sort -u z1.list>z.list
rm z1.list
rm MARCKSL1.snps

for (( c=1; c<=$(cat z.list|wc -l); c++ ))
do
z=$(awk -v var=$c '{ if (NR==var) print $0 }' z.list)
awk -v var=$z '{ if (NR==var) print $0 }' MARCKSL1.TESTS>>MARCKSL1.snps
done


fgrep -wf MARCKSL1.snps ./Simulation/hapmap.out.controls.gen |awk '{print
$3}'>MARCKSL1.list


###Stop Here###
fgrep -wf MARCKSL1.list ./Simulation/p3.assoc

awk '{ if (NR==1) print $0 " " 1" " 1.5 " " 1.5*1.5}' MARCKSL1.list>>MARCKSL1.dl
awk '{ if (NR==2) print $0 " " 1" " 1.6 " " 1.6*1.6}' MARCKSL1.list>>MARCKSL1.dl
awk '{ if (NR==3) print $0 " " 1" " 1.7 " " 1.7*1.7}' MARCKSL1.list>>MARCKSL1.dl


cat MARCKSL1.dl|tr "\n" " ">list.txt
rm MARCKSL1.dl
cat MARCKSL1.snps>fox.txt
cat fox.txt>MARCKSL1.snps
cat list.txt>MARCKSL1.effects
```

```
rm ./Simulation/p3.assoc

###Simulation###
fgrep -w MARCKSL1 genelist1.txt|awk '{print $1}'>snplist.txt
z=$(head -n 1 snplist.txt)
z1=$(tail -n 1 snplist.txt)

for i in {1..10}
do
t=$(cat MARCKSL1.effects)
m=0
n=0
c=3
f=2
v=0
r=1
while [ $c -gt $f ]
do
while [ $m -eq $n ]
do
./hapgen2 -m ./CEU.0908.impute.files/MARCKSL1.map -l
./CEU.0908.impute.files/MARCKSL1.legend -h ./CEU.0908.impute.files/MARCKSL1.hap -o
./Simulation/single$i.out  -no_haps_output -dl $t -n 1000 1000|fgrep "seed"|awk '{print
$5}'>c.txt
cat c.txt>>seed.txt
m=$(cat c.txt|tr -d '.' |tr -d 'e' |tr -d '-')
done
n=$m
./gtool -G --g ./Simulation/single$i.out.cases.gen --s ./Simulation/single$i.out.cases.sample --ped
./Simulation/out.ped --map ./Simulation/out.map
./gtool -G --g ./Simulation/single$i.out.controls.gen --s ./Simulation/single$i.out.controls.sample
--ped ./Simulation/out1.ped --map ./Simulation/out1.map
awk '{print $1 " " $2 " " $3 " " $4 " " 1 " " 2}' ./Simulation/out.ped>./Simulation/outa.ped
cat ./Simulation/out.ped|tr '\t' ' '|cut -d ' ' -f7- >./Simulation/outb.ped
paste ./Simulation/outa.ped ./Simulation/outb.ped>./Simulation/out2.ped
cat ./Simulation/out.map>./Simulation/out4.map
awk  -v var=$i '{print 1 " " $2 " " 0 " " $4}' ./Simulation/out.map>./Simulation/out4.map
awk '{print $1 " " $2 " " $3 " " $4 " " 1 " " 1}' ./Simulation/out1.ped>./Simulation/out3a.ped
cat ./Simulation/out1.ped|tr '\t' ' '|cut -d ' ' -f7- >./Simulation/out3b.ped
paste ./Simulation/out3a.ped ./Simulation/out3b.ped>./Simulation/out3.ped
cat ./Simulation/out2.ped ./Simulation/out3.ped>./Simulation/out4.ped
cat ./Simulation/out4.ped>./Simulation/b.ped
cat ./Simulation/out4.map>./Simulation/b.map
x=$(fgrep -w $(fgrep -w $(echo $z) plink.assoc|awk 'NR==1{print $3}')
./Simulation/out4.map|awk '{print $2}')
```

```
x1=$(fgrep -w $(fgrep -w $(echo $z1) plink.assoc|awk 'NR==1{print $3}')
./Simulation/out4.map|awk '{print $2}')
plink --file ./Simulation/b --recode --from $(echo $x) --to $(echo $x1) --out ./Simulation/pop$i --
noweb
plink --file ./Simulation/pop$i --logistic --extract MARCKSL1.snps --noweb
plink --file ./Simulation/pop$i --logistic --noweb --maf .05 --hwe    .0001 --from $(echo $x) --to
$(echo $x1)
r=$(awk '{if ($9<.00001) print $0}' plink.assoc.logistic|wc -l)
if [ $v -lt $r ]; then
f=0
else
f=$(fgrep -wf MARCKSL1.snps plink.assoc.logistic|awk '{if ($9>.00001 && $9<.0005) print
$0}'|wc -l)
fi
fgrep -wf MARCKSL1.snps plink.assoc.logistic
rm ./Simulation/plink.assoc.logistic
rm ./Simulation/sample*
rm ./Simulation/*.summary
rm ./Simulation/single*
rm ./Simulation/pop*.log
done
done
zip MARCKSL1.zip ./Simulation/pop*

###Gene Plot###
###pop indicates one of 1000 data sets for gene###
###Made for gene SYPL2###
###Create a matrix of all p-values for each marker###
plink --file pop1 --maf .05 --hwe .0001 --logistic --noweb
awk '{print $2  "," $NF }' plink.assoc.logistic|sed '1,1d'|sort>p.txt

for i in {2..10};
do
plink --file pop$i --logistic --maf .05 --hwe .0001
awk '{print $2 "," $NF}' plink.assoc.logistic|sed '1,1d'|sort>p1.txt
join -t, -a1 -a2 p.txt p1.txt>p2.txt
cat p2.txt>p.txt
done

for i in {1..10};
do
cat pop$i.map>>all.map
done

sort -u all.map>all1.map
```

```
cat p.txt|cut -d ',' -f2->p1.txt

awk -F, '{print NF}' p1.txt>s.txt
paste s.txt p.txt|sort -r -n|cut -f2->pvalue.csv

###R-code to make plot###
setwd("C:/Users/Harrison/Documents/Genebasedmethod/Results/Newgenes/SYPL2/Simulation"
)
r<-read.csv("pvalue.csv",header=FALSE)
n<-dim(r)[2]
b <- sapply(r[,2:n],function(x)as.numeric(as.character(x)))
ap<-apply(b,1,median,na.rm=TRUE)
###install.packages("ggplot2")###
library(ggplot2)
###Manhattan plot from Getting Genetic Done Blog###
manhattan <- function(dataframe, colors=c("gray10", "gray50"), ymax="max",
limitchromosomes=1:23, suggestiveline=0, genomewideline=0,
annotate=NULL,annotate1=NULL,...) {

   d=dataframe
   if (!("CHR" %in% names(d) & "BP" %in% names(d) & "P" %in% names(d))) stop("Make
sure your data frame contains columns CHR, BP, and P")

   if (any(limitchromosomes)) d=d[d$CHR %in% limitchromosomes, ]
   d=subset(na.omit(d[order(d$CHR, d$BP), ]), (P>0 & P<=1)) # remove na's, sort, and keep
only 0<P<=1
   d$logp = -log10(d$P)
   d$pos=NA
   ticks=NULL
   lastbase=0
   colors <- rep(colors,max(d$CHR))[1:max(d$CHR)]
   if (ymax=="max") ymax<-ceiling(max(d$logp))
   if (ymax<5) ymax<-5

   numchroms=length(unique(d$CHR))
   if (numchroms==1) {
      d$pos=d$BP
      ticks=floor(length(d$pos))/2+1
   } else {
      for (i in unique(d$CHR)) {
        if (i==1) {
                        d[d$CHR==i, ]$pos=d[d$CHR==i, ]$BP
                } else {
                        lastbase=lastbase+tail(subset(d,CHR==i-1)$BP, 1)
                        d[d$CHR==i, ]$pos=d[d$CHR==i, ]$BP+lastbase
                }
```

```
                ticks=c(ticks, d[d$CHR==i, ]$pos[floor(length(d[d$CHR==i, ]$pos)/2)+1])
        }
    }

    if (numchroms==1) {
        with(d, plot(pos, logp, ylim=c(0,ymax), ylab=expression(-log[10](italic(p))),
xlab=paste("Chromosome",unique(d$CHR),"position"),...))
        }       else {
        with(d, plot(pos, logp, ylim=c(0,ymax), ylab=expression(-log[10](italic(p))),
xlab="Chromosome", xaxt="n", type="n", ...))
        axis(1, at=ticks, lab=unique(d$CHR), ...)
        icol=1
        for (i in unique(d$CHR)) {
            with(d[d$CHR==i, ],points(pos, logp, col=colors[icol], ...))
            icol=icol+1
            }
    }

    if (!is.null(annotate)) {
        d.annotate=d[which(d$SNP %in% annotate), ]
        with(d.annotate, points(pos, logp, col="blue3", ...))
    }
     if (!is.null(annotate1)) {
        d.annotate=d[which(d$SNP %in% annotate1), ]
        with(d.annotate, points(pos, logp, col="red3", ...))
    }
    if (suggestiveline) abline(h=suggestiveline, col="blue")
    if (genomewideline) abline(h=genomewideline, col="red")
}

r1<-read.table("all1.map")
r2<-cbind(r[,1:2],ap)
library(gdata)
colnames(r2)<-c("V2","V","P")
m<-merge(trim(r1),trim(r2),by="V2")
r3<-cbind(m[,1:2],1,m[,4],m[,6])
r3<-r3[,-2]
colnames(r3)<-c("SNP","CHR","BP","P")
l1<-c(35793880,35809319,35840445)
f<-as.character(r1[r1[,4]%in%as.numeric(l1),2])
jpeg(file="SYPL2.jpeg")
manhattan(data.frame(r3),annotate1=f)
title("SYPL2 Gene Plot")
dev.off()
```

4.2.2.1 Necessity of Permutation

```
###Q-Q plot###
QQ plot taken from getting genetics done blog for use in R
http://gettinggeneticsdone.blogspot.com/2011/04/annotated-manhattan-plots-and-qq-plots.html
###code source###
source("http://people.virginia.edu/~sdt5z/0STABLE/qqman.r")
```

4.2.2.3 Power

```
###power.csv contains p-values for each method###
###code written for R###
r<-read.csv("power.csv",header=FALSE)
plot(r[which(r[,1]=="Gabriel"),4],type="o", col="blue", ylim=c(0,1),axes=FALSE,ann=FALSE )
axis(2, las=1, at=c(0,.25,.5,.75,1))
box()
lines(r[which(r[,1]=="HapBlock"),4], type="o", pch=21, lty=1,col="red")
lines(r[which(r[,1]=="GWIS"),4], type="o", pch=21, lty=1,col="orange")
lines(r[which(r[,1]=="Fisher"),4], type="o", pch=21, lty=1,col="purple")
lines(r[which(r[,1]=="Vegas"),4], type="o", pch=21, lty=1,col="black")
title(xlab= "Genes")
title(ylab= "Power")
axis(1, at=1:6, lab=c("SYPL2", "C1orf92", "FOXE3", "CACHD1", "TMEM51","HS2ST1"))
legend("bottomright",1,unique(r[,1]), cex=0.8, col=c("red","orange","purple","black"),pch=21,
lty=1);
```

4.3.2 Small P-value Power Assessment Results

4.3.2.1 Power Plot Code

```
##Power Plot###
###SummarySE taken from cookbook for r###
###http://wiki.stdout.org/rcookbook/Manipulating%20data/Summarizing%20data/###
summarySE <- function(data=NULL, measurevar, groupvars=NULL, na.rm=FALSE,
conf.interval=.95, .drop=TRUE) {
   require(plyr)

   # New version of length which can handle NA's: if na.rm==T, don't count them
   length2 <- function (x, na.rm=FALSE) {
      if (na.rm) sum(!is.na(x))
      else       length(x)
   }

   # This is does the summary; it's not easy to understand...
   datac <- ddply(data, groupvars, .drop=.drop,
               .fun= function(xx, col, na.rm) {
                     c( N    = length2(xx[,col], na.rm=na.rm),
                       mean = mean   (xx[,col], na.rm=na.rm),
                       sd   = sd     (xx[,col], na.rm=na.rm)
                       )
```

```
                },
            measurevar,
            na.rm
        )

  # Rename the "mean" column
  datac <- rename(datac, c("mean"=measurevar))

  datac$se <- datac$sd / sqrt(datac$N)  # Calculate standard error of the mean

  # Confidence interval multiplier for standard error
  # Calculate t-statistic for confidence interval:
  # e.g., if conf.interval is .95, use .975 (above/below), and use df=N-1
  ciMult <- qt(conf.interval/2 + .5, datac$N-1)
  datac$ci <- datac$se * ciMult

  return(datac)
}


r1<-read.csv("correlationplot1.csv")

colnames(r1)<-c("Method","Gene","-log10(P)")

library(ggplot2)
dfc <- summarySE(r1, measurevar="-log10(P)", groupvars=c("Method","Gene"))

pd <- position_dodge(.5)
ggplot(dfc, aes(x=Gene, y=dfc[,4], colour=Method)) +
  geom_errorbar(aes(ymin=dfc[,4]-ci, ymax=dfc[,4]+ci), width=.1, position=pd) +
      geom_line(position=pd) +
  geom_point(position=pd)+ ylab("-log10(P)")
```

4.3.2.1 Correlation Plot

```
##correlation plot is the specified format for pairs command###
linear<-function(x,y){
 points(x,y)
 res=lm(y~x)
 abline(res)
 }

panel.cor <- function(x, y, digits=2, prefix="", cex.cor)
{
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
```

```
    r <- abs(cor(x, y))
    txt <- format(c(r, 0.123456789), digits=digits)[1]
    txt <- paste(prefix, txt, sep="")
    if(missing(cex.cor)) cex <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex)
}

r<-read.csv("correlationplot.csv")

pairs(r, lower.panel=linear, upper.panel=panel.cor)
```

4.3.2.2 Comparison of Blocking Methods

```
##Counts represent the number of blocks of a give size for Gabriel(G) and Hapblock Method###
##R-Code##
r<-read.table("countG.txt")
r1<-read.table("countH.txt")
h<-hist(as.matrix(r[which(r<26),]))
h1<-hist(as.matrix(r1[which(r1<26),]))
ha<-hist(as.matrix(r[which(r>25),],))
h1a<-hist(as.matrix(r1[which(r1>25),]),breaks=seq(25,80,by=5))
plot( h, col=rgb(0,0,1,1/4),xlim=c(0,25),ylim=c(0,1000))  # first histogram
plot( h1, col=rgb(1,0,0,1/4), xlim=c(0,25),add=T)

jpeg(file = "blockcompare25.jpeg")
plot( h, col=rgb(0,0,1,1/4),xlim=c(0,25),ylim=c(0,1000),main="Gabriel vs HapBlock Block Size
<= 25",xlab="Block Size (#SNPs)")  # first histogram
plot( h1, col=rgb(1,0,0,1/4), xlim=c(0,25),add=T)
legend('topright',c('HapBlock','Gabriel'), fill = rgb(1:0,0,0:1,0.4), bty = 'n', border = NA)
dev.off()

jpeg(file = "blockcompare80.jpeg")
plot( ha, col=rgb(0,0,1,1/4),xlim=c(25,80),ylim=c(0,25),main="Gabriel vs HapBlock Block Size
>25",xlab="Block Size (#SNPs)")  # first histogram
plot( h1a, col=rgb(1,0,0,1/4),xlim=c(25,80),add=T)
legend('topright',c('HapBlock','Gabriel'), fill = rgb(1:0,0,0:1,0.4), bty = 'n', border = NA)
dev.off()
```

5.0 Gene-Based Testing of Alzheimer's GWAS

5.1.5 Imputation

```
###Example Chromosome###
###Frank Submit###
#!/bin/bash
#PBS -m e
#PBS -M hab45@pitt.edu
#PBS -j oe
```

```
#PBS -q shared
#PBS -N impute1
#PBS -l nodes=1:ppn=1
#PBS -l walltime=44:00:00
#PBS -l vmem=2GB
#PBS -S /bin/bash
cd /home/bdiergaarde/hab45/GWIS/1.1

###Plink command; Merged data set is comprised of ALZ GWAS and additionally all controls coded to be imputation reference ###
plink --bfile merged --proxy-impute all --make-bed --out imputed1 --noweb --chr 1 --proxy-impute-threshold 0
```

5.1.6.1 Standard Single-SNP GWAS Analysis

```
###Single SNP analysis##
###plink command##
plink --bfile alz --hwe .000001 –maf .05
###Manhattan and Q-Q Plot###
##From getting Genetics Done Blog##
## http://gettinggeneticsdone.blogspot.com/2011/04/annotated-manhattan-plots-and-qq-plots.html###
setwd("/Users/hab45/Desktop/GWIS/ALZ")
source("http://people.virginia.edu/~sdt5z/0STABLE/qqman.r")
r<-read.csv("noapoe.csv",header=TRUE)
###change really large pvalues to 1E-10 so it graphs nicer###
jpeg('manhattan.jpg')
dev.new(width=23, height=5)
manhattan(r,colors=c("black","#666666","#CC6600"),genomewideline=F,suggestiveline=2.38e-7)
dev.off()
jpeg('qq.jpg')
qq(r$P)
dev.off()
```

5.1.6.2 Haplotype Analysis

```
###On Frank##
###For all chromosome###
for i in {1..22}
do
echo " #!/bin/bash
#PBS -m e
#PBS -M hab45@pitt.edu
#PBS -j oe
#PBS -q shared
#PBS -N gwis$i
```

```
#PBS -l nodes=1:ppn=1
#PBS -l walltime=23:00:00
#PBS -l vmem=4GB
#PBS -S /bin/bash
cd /home/bdiergaarde/hab45/GWIS/

plink --bfile ./1.1/data1 --blocks --chr $i --out block$i
plink --bfile ./1.1/data1 --hap block$i.blocks --hap-logistic --hap-omnibus --out chr$i
plink --bfile ./1.1/data1 --hap block$i.blocks --hap-logistic --hap-omnibus --out hap$i
">blocks$i.txt
done

##Submit###
for i in {1..22}
do
qsub blocks$i.txt
done
```

5.1.6.3 Gene Level Analysis

```
###Annotation of GWAS data; example of chromosome $1###
#! /bin/bash
chmod a+x plink.exe
./plink.exe --bfile data --assoc --chr $1
./plink.exe --annotate plink.assoc ranges=glist.txt --border 25 --out plink$1
awk '{print $2}' plink$1.annot|sed '1,1d'>snps.txt
awk '{print $NF}' plink$1.annot|sed '1,1d'>genelist.txt
sed -e 's/([^()]*)//g' genelist.txt |tr '|' ' '>genel.txt
paste -d " " snps.txt genel.txt>genelist$1.txt
cut -d ' ' -f2- genelist$1.txt>g.txt
tr ' ' '\n' <g.txt|sort -u|fgrep -wv .>genes$1.txt
rm genelist.txt
cat genelist*.txt>all.genelist
cat genes*.txt>all.gene.txt
c=$(wc -l all.gene.txt|awk '{print $1/1000}'|awk '{printf "%.0f\n", $1}')
split -a 3 --lines=$c all.gene.txt ./genelist/x
cd genelist
ls -l x*|awk '{print $NF}'>names.txt
cd ..
for (( c=1; c<=$(cat ./genelist/names.txt|wc -l); c++ ))
do
i=$(awk -v var=$c 'NR==var{print $0}' ./genelist/names.txt)
fgrep -wf ./genelist/$i all.genelist|awk '{print $1}'>./snplist/snplist$i.txt
done
for (( c=1; c<=$(cat ./genelist/names.txt|wc -l); c++ ))
```

```
do
i=$(awk -v var=$c 'NR==var{print $0}' ./genelist/names.txt)
for (( k=1; k<=$(cat ./genelist/$i|wc -l); k++ ))
do
j=$(awk -v var=$k 'NR==var{print $0}' ./genelist/$i)
fgrep -w $j all.genelist|awk -v var=$j '{print var " " $1}'>>./snplist/genesnps$i.txt
done
done
###Any gene name can be subsisted for example###
plink --bfile data --recode --extract snplistexample.txt --out example

###See above sections for each gene-based methods code###
###Modified Q-Q plot with each gene written in R###
###Combine p-values from each gene based method###
r<-read.csv("vegasfinal.csv",header=TRUE)
r1<-read.table("fishernoAPOE.txt")
r2<-read.table("gabrielall.txt")
r3<-read.table("hapblockall.p")
r4<-read.table("gwisalzresultsqq.txt")
m<-m <- matrix(seq(1:17547))
r5<-apply(as.matrix(1:17547),1,function(x) x/17547)

jpeg('qqall.jpg')
qqplot(-log10(r5[1:length(sort(r2[,5]))]),-log10(sort(r2[,5])),xlab="Expected -
log10(p)",ylab="Observed -log10(p)")
abline(0,1)
points(-log10(r5),-log10(sort(r1[,5])),col="blue")
points(-log10(r5),-log10(sort(r[,8])),col="red")
points(-log10(r5[1:length(sort(r3[,5]))]),-log10(sort(r3[,5])),col="green4")
points(-log10(r5[1:length(sort(r4[,20]))]),-log10(sort(r4[,20])),col="purple")
legend("topleft", c("GeneBlock-Gabriel","GeneBlock-HapBlock",
"Vegas","Fisher","GWiS"),pch=1 ,col=c("black","green4","red","blue","purple"))
dev.off()
```

BIBLIOGRAPHY

Achkar JP, Klei L, de Bakker PI, Bellone G, Rebert N, Scott R, Lu Y, Regueiro M, Brzezinski A, Kamboh MI and others. 2012. Amino acid position 11 of HLA-DRbeta1 is a major determinant of chromosome 6p association with ulcerative colitis. Genes Immun 13(3):245-52.

AlzheimerAssociation. 2012. 2012 Alzheimer's disease facts and figures. Alzheimers Dement 8(2):131-168.

Anderson EC, Novembre J. 2003. Finding haplotype block boundaries by using the minimum-description-length principle. Am J Hum Genet 73(2):336-54.

Asimit J, Zeggini E. 2010. Rare variant association analysis methods for complex traits. Annu Rev Genet 44:293-308.

Baba Y, Iida A, Watanabe S. 2011. Sall3 plays essential roles in horizontal cell maturation through regulation of neurofilament expression levels. Biochimie 93(6):1037-46.

Bacanu SA. 2012. On optimal gene-based analysis of genome scans. Genet Epidemiol 36(4):333-9.

Ballard C, Gauthier S, Corbett A, Brayne C, Aarsland D, Jones E. 2011. Alzheimer's disease. Lancet 377(9770):1019-31.

Barrett JC. 2009. Haploview: Visualization and analysis of SNP genotype data. Cold Spring Harb Protoc 2009(10):pdb ip71.

Barrett JC, Fry B, Maller J, Daly MJ. 2005. Haploview: analysis and visualization of LD and haplotype maps. Bioinformatics 21(2):263-5.

Berger D. 2006. A Gentle Introduction to Resampling Techniques. Western Psychological Association Statistical Workshop.

Bertram L, Tanzi RE. 2009. Genome-wide association studies in Alzheimer's disease. Hum Mol Genet 18(R2):R137-45.

Besag J, Clifford P. 1991. Sequential Monte Carlo p-values. Biometrika 78(2):301-304.

Bonferroni CE. 1935. Il calcolo delle assicurazioni su gruppi di teste: Tipografia del Senato.

Broman KW, Caffo BS. 2003. Simulation-based P values: response to North et al. Am J Hum Genet 72(2):496.

Browning SR, Browning BL. 2011. Haplotype phasing: existing methods and new developments. Nat Rev Genet 12(10):703-14.

Buil A, Martinez-Perez A, Perera-Lluna A, Rib L, Caminal P, Soria JM. 2009. A new gene-based association test for genome-wide association studies. BMC Proc 3 Suppl 7:S130.

Bush WS, Chen G, Torstenson ES, Ritchie MD. 2009. LD-spline: mapping SNPs on genotyping platforms to genomic regions using patterns of linkage disequilibrium. BioData Min 2(1):7.

Cervantes S, Samaranch L, Vidal-Taboada JM, Lamet I, Bullido MJ, Frank-Garcia A, Coria F, Lleo A, Clarimon J, Lorenzo E and others. 2011. Genetic variation in APOE cluster region and Alzheimer's disease risk. Neurobiol Aging 32(11):2107 e7-17.

Chapman JM, Cooper JD, Todd JA, Clayton DG. 2003. Detecting disease associations due to linkage disequilibrium using haplotype tags: a class of tests and the determinants of statistical power. Hum Hered 56(1-3):18-31.

Choi M, Scholl UI, Ji W, Liu T, Tikhonova IR, Zumbo P, Nayir A, Bakkaloglu A, Ozen S, Sanjad S and others. 2009. Genetic diagnosis by whole exome capture and massively parallel DNA sequencing. Proc Natl Acad Sci U S A 106(45):19096-101.
Chuikov S, Levi BP, Smith ML, Morrison SJ. 2010. Prdm16 promotes stem cell maintenance in multiple tissues, partly by regulating oxidative stress. Nat Cell Biol 12(10):999-1006.
Chung SJ, Lee JH, Kim SY, You S, Kim MJ, Lee JY, Koh J. 2012. Association of GWAS Top Hits With Late-onset Alzheimer Disease in Korean Population. Alzheimer Dis Assoc Disord.
Clayton DG. 2009. Sex chromosomes and genetic association studies. Genome Med 1(11):110.
Consortium IH. 2003. The International HapMap Project. Nature 426(6968):789-96.
Coon KD, Myers AJ, Craig DW, Webster JA, Pearson JV, Lince DH, Zismann VL, Beach TG, Leung D, Bryden L and others. 2007. A high-density whole-genome association study reveals that APOE is the major susceptibility gene for sporadic late-onset Alzheimer's disease. J Clin Psychiatry 68(4):613-8.
Culpan D, Cram D, Chalmers K, Cornish A, Palmer L, Palmer J, Hughes A, Passmore P, Craig D, Wilcock GK and others. 2009. TNFR-associated factor-2 (TRAF-2) in Alzheimer's disease. Neurobiol Aging 30(7):1052-60.
Curtis D, Vine AE, Knight J. 2008. A simple method for assessing the strength of evidence for association at the level of the whole gene. Adv Appl Bioinform Chem 1:115-20.
Dagmar EE, Bibiana KYW, Michael RH. 2011. Convergent pathogenic pathways in Alzheimer's and Huntington's diseases: shared targets for drug development. Nature Reviews Drug Discovery 10(11):853-867.
de Bakker PI, Yelensky R, Pe'er I, Gabriel SB, Daly MJ, Altshuler D. 2005. Efficiency and power in genetic association studies. Nat Genet 37(11):1217-23.
De La Vega FM, Bustamante CD, Leal SM. 2011. Genome-Wide Association Mapping and Rare Alleles: From Population Genomics to Personalized Medicine - Session Introduction. Pac Symp Biocomput:74-5.
Dennissen FJ, Kholod N, van Leeuwen FW. 2012. The ubiquitin proteasome system in neurodegenerative diseases: culprit, accomplice or victim? Prog Neurobiol 96(2):190-207.
Derkach A, Lawless JF, Sun L. 2012. Robust and Powerful Tests for Rare Variants Using Fisher's Method to Combine Evidence of Association From Two or More Complementary Tests. Genet Epidemiol.
Duggal P, Gillanders EM, Holmes TN, Bailey-Wilson JE. 2008. Establishing an adjusted p-value threshold to control the family-wide type 1 error in genome wide association studies. BMC Genomics 9:516.
Eichler EE, Flint J, Gibson G, Kong A, Leal SM, Moore JH, Nadeau JH. 2010. Missing heritability and strategies for finding the underlying causes of complex disease. Nat Rev Genet 11(6):446-50.
Estey MP, Kim MS, Trimble WS. 2011. Septins. Curr Biol 21(10):R384-7.
Evans DA, Funkenstein HH, Albert MS, Scherr PA, Cook NR, Chown MJ, Hebert LE, Hennekens CH, Taylor JO. 1989. Prevalence of Alzheimer's disease in a community population of older persons. Higher than previously reported. JAMA 262(18):2551-6.
Ewens WJ. 2003. On estimating P values by the Monte Carlo method. Am J Hum Genet 72(2):496-8.

Fallin D, Schork NJ. 2000. Accuracy of haplotype frequency estimation for biallelic loci, via the expectation-maximization algorithm for unphased diploid genotype data. Am J Hum Genet 67(4):947-59.
Ferri CP, Prince M, Brayne C, Brodaty H, Fratiglioni L, Ganguli M, Hall K, Hasegawa K, Hendrie H, Huang Y and others. 2005. Global prevalence of dementia: a Delphi consensus study. Lancet 366(9503):2112-7.
Fisher RA. 1925. Statistical methods for research workers 13 ed: Oliver and Boyd.
Flach H, Rosenbaum M, Duchniewicz M, Kim S, Zhang SL, Cahalan MD, Mittler G, Grosschedl R. 2010. Mzb1 protein regulates calcium homeostasis, antibody secretion, and integrin activation in innate-like B cells. Immunity 33(5):723-35.
Fratiglioni L, De Ronchi D, Aguero-Torres H. 1999. Worldwide prevalence and incidence of dementia. Drugs Aging 15(5):365-75.
Fruhbeck G, Sesma P, Burrell MA. 2009. PRDM16: the interconvertible adipo-myocyte switch. Trends Cell Biol 19(4):141-6.
Fujita PA, Rhead B, Zweig AS, Hinrichs AS, Karolchik D, Cline MS, Goldman M, Barber GP, Clawson H, Coelho A and others. 2011. The UCSC Genome Browser database: update 2011. Nucleic Acids Res 39(Database issue):D876-82.
Furujo M, Kinoshita M, Nagao M, Kubo T. 2012. Methionine adenosyltransferase I/III deficiency: Neurological manifestations and relevance of S-adenosylmethionine. Mol Genet Metab 107(3):253-6.
Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M and others. 2002. The structure of haplotype blocks in the human genome. Science 296(5576):2225-9.
Gao Q, He Y, Yuan Z, Zhao J, Zhang B, Xue F. 2011. Gene- or region-based association study via kernel principal component analysis. BMC Genet 12:75.
Gatz M, Pedersen NL, Berg S, Johansson B, Johansson K, Mortimer JA, Posner SF, Viitanen M, Winblad B, Ahlbom A. 1997. Heritability for Alzheimer's disease: the study of dementia in Swedish twins. J Gerontol A Biol Sci Med Sci 52(2):M117-25.
Gauderman WJ, Murcray C, Gilliland F, Conti DV. 2007. Testing association between disease and multiple SNPs in a candidate gene. Genet Epidemiol 31(5):383-95.
Grisanzio C, Freedman ML. 2010. Chromosome 8q24-Associated Cancers and MYC. Genes Cancer 1(6):555-9.
Hamza TH, Zabetian CP, Tenesa A, Laederach A, Montimurro J, Yearout D, Kay DM, Doheny KF, Paschall J, Pugh E and others. 2010. Common genetic variation in the HLA region is associated with late-onset sporadic Parkinson's disease. Nat Genet 42(9):781-5.
HDCRG. 1993. A novel gene containing a trinucleotide repeat that is expanded and unstable on Huntington's disease chromosomes. The Huntington's Disease Collaborative Research Group. Cell 72(6):971-83.
He Y, Li C, Amos CI, Xiong M, Ling H, Jin L. 2011. Accelerating haplotype-based genome-wide association study using perfect phylogeny and phase-known reference data. PLoS One 6(7):e22097.
Hebert LE, Scherr PA, Bienias JL, Bennett DA, Evans DA. 2003. Alzheimer disease in the US population: prevalence estimates using the 2000 census. Arch Neurol 60(8):1119-22.
Hibar DP, Stein JL, Kohannim O, Jahanshad N, Saykin AJ, Shen L, Kim S, Pankratz N, Foroud T, Huentelman MJ and others. 2011. Voxelwise gene-wide association study

(vGeneWAS): multivariate gene-based association testing in 731 elderly subjects. Neuroimage 56(4):1875-91.
Hill-Burns EM, Factor SA, Zabetian CP, Thomson G, Payami H. 2011. Evidence for more than one Parkinson's disease-associated variant within the HLA region. PLoS One 6(11):e27109.
Hindorff LA, Sethupathy P, Junkins HA, Ramos EM, Mehta JP, Collins FS, Manolio TA. 2009. Potential etiologic and functional implications of genome-wide association loci for human diseases and traits. Proc Natl Acad Sci U S A 106(23):9362-7.
Hong MG, Alexeyenko A, Lambert JC, Amouyel P, Prince JA. 2010. Genome-wide pathway analysis implicates intracellular transmembrane protein transport in Alzheimer disease. J Hum Genet 55(10):707-9.
Howie B, Fuchsberger C, Stephens M, Marchini J, Abecasis GR. 2012. Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. Nat Genet 44(8):955-9.
Huang H, Chanda P, Alonso A, Bader JS, Arking DE. 2011. Gene-based tests of association. PLoS Genet 7(7):e1002177.
Idriss HT, Naismith JH. 2000. TNF alpha and the TNF receptor superfamily: structure-function relationship(s). Microsc Res Tech 50(3):184-95.
Indap AR, Marth GT, Struble CA, Tonellato P, Olivier M. 2005. Analysis of concordance of different haplotype block partitioning algorithms. BMC Bioinformatics 6:303.
Johnson RC, Nelson GW, Troyer JL, Lautenberger JA, Kessing BD, Winkler CA, O'Brien SJ. 2010. Accounting for multiple comparisons in a genome-wide association study (GWAS). BMC Genomics 11:724.
Joo HY, Jones A, Yang C, Zhai L, Smith ADt, Zhang Z, Chandrasekharan MB, Sun ZW, Renfrow MB, Wang Y and others. 2011. Regulation of histone H2A and H2B deubiquitination and Xenopus development by USP12 and USP46. J Biol Chem 286(9):7190-201.
Kamboh MI, Barmada MM, Demirci FY, Minster RL, Carrasquillo MM, Pankratz VS, Younkin SG, Saykin AJ, Sweet RA, Feingold E and others. 2011. Genome-wide association analysis of age-at-onset in Alzheimer's disease. Mol Psychiatry.
Kamboh MI, Barmada MM, Demirci FY, Minster RL, Carrasquillo MM, Pankratz VS, Younkin SG, Saykin AJ, Sweet RA, Feingold E and others. 2012a. Genome-wide association analysis of age-at-onset in Alzheimer's disease. Mol Psychiatry 17(12):1340-1346.
Kamboh MI, Demirci FY, Wang X, Minster RL, Carrasquillo MM, Pankratz VS, Younkin SG, Saykin AJ, Jun G, Baldwin C and others. 2012b. Genome-wide association study of Alzheimer's disease. Transl Psychiatry 2:e117.
Kanehisa M, Goto S. 2000. KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res 28(1):27-30.
Khatri P, Sirota M, Butte AJ. 2012. Ten years of pathway analysis: current approaches and outstanding challenges. PLoS Comput Biol 8(2):e1002375.
Kinnamon DD, Hershberger RE, Martin ER. 2012. Reconsidering association testing methods using single-variant test statistics as alternatives to pooling tests for sequence data with rare variants. PLoS One 7(2):e30238.
Kohlhase J, Hausmann S, Stojmenovic G, Dixkens C, Bink K, Schulz-Schaeffer W, Altmann M, Engel W. 1999. SALL3, a new member of the human spalt-like gene family, maps to 18q23. Genomics 62(2):216-22.

Kuhlenbaumer G, Hannibal MC, Nelis E, Schirmacher A, Verpoorten N, Meuleman J, Watts GD, De Vriendt E, Young P, Stogbauer F and others. 2005. Mutations in SEPT9 cause hereditary neuralgic amyotrophy. Nat Genet 37(10):1044-6.

Kumar U. 2005. Expression of somatostatin receptor subtypes (SSTR1-5) in Alzheimer's disease brain: an immunohistochemical analysis. Neuroscience 134(2):525-38.

Lambert JC, Grenier-Boley B, Chouraki V, Heath S, Zelenika D, Fievet N, Hannequin D, Pasquier F, Hanon O, Brice A and others. 2010. Implication of the immune system in Alzheimer's disease: evidence from genome-wide pathway analysis. J Alzheimers Dis 20(4):1107-18.

Li H. 2012. U-statistics in genetic association studies. Hum Genet 131(9):1395-401.

Li H, Wetten S, Li L, St Jean PL, Upmanyu R, Surh L, Hosford D, Barnes MR, Briley JD, Borrie M and others. 2008. Candidate single-nucleotide polymorphisms from a genomewide association study of Alzheimer disease. Arch Neurol 65(1):45-53.

Li J. 2008. A novel strategy for detecting multiple loci in Genome-Wide Association Studies of complex diseases. Int J Bioinform Res Appl 4(2):150-63.

Li MX, Gui HS, Kwan JS, Sham PC. 2011. GATES: a rapid and powerful gene-based association test using extended Simes procedure. Am J Hum Genet 88(3):283-93.

Li MX, Kwan JS, Sham PC. 2012. HYST: A Hybrid Set-Based Test for Genome-wide Association Studies, with Application to Protein-Protein Interaction-Based Association Analysis. Am J Hum Genet 91(3):478-88.

Libioulle C, Louis E, Hansoul S, Sandor C, Farnir F, Franchimont D, Vermeire S, Dewit O, de Vos M, Dixon A and others. 2007. Novel Crohn disease locus identified by genome-wide association maps to a gene desert on 5p13.1 and modulates expression of PTGER4. PLoS Genet 3(4):e58.

Liptak T. 1958. On the combination of independent tests. Magyar Tud. Akad. Mat. Kutato Int.(3):171–197.

Liu JZ, McRae AF, Nyholt DR, Medland SE, Wray NR, Brown KM, Hayward NK, Montgomery GW, Visscher PM, Martin NG and others. 2010. A versatile gene-based test for genome-wide association studies. Am J Hum Genet 87(1):139-45.

Liu N, Zhang K, Zhao H. 2008. Haplotype-association analysis. Adv Genet 60:335-405.

Logue MW, Schu M, Vardarajan BN, Buros J, Green RC, Go RC, Griffith P, Obisesan TO, Shatz R, Borenstein A and others. 2011. A comprehensive genetic association study of Alzheimer disease in African Americans. Arch Neurol 68(12):1569-79.

Long JC, Williams RC, Urbanek M. 1995. An E-M algorithm and testing strategy for multiple-locus haplotypes. Am J Hum Genet 56(3):799-810.

Lorenz AJ, Hamblin MT, Jannink JL. 2010. Performance of single nucleotide polymorphisms versus haplotypes for genome-wide association analysis in barley. PLoS One 5(11):e14079.

Madsen BE, Browning SR. 2009. A groupwise association test for rare mutations using a weighted sum statistic. PLoS Genet 5(2):e1000384.

Manolio TA, Collins FS, Cox NJ, Goldstein DB, Hindorff LA, Hunter DJ, McCarthy MI, Ramos EM, Cardon LR, Chakravarti A and others. 2009. Finding the missing heritability of complex diseases. Nature 461(7265):747-53.

Marchini J, Cutler D, Patterson N, Stephens M, Eskin E, Halperin E, Lin S, Qin ZS, Munro HM, Abecasis GR and others. 2006. A comparison of phasing algorithms for trios and unrelated individuals. Am J Hum Genet 78(3):437-50.

Marchini J, Howie B. 2010. Genotype imputation for genome-wide association studies. Nat Rev Genet 11(7):499-511.
Mardia K. 1975. Assessment of multinormality and the robustness of Hotelling's T 2 test. Applied Statistics:163-171.
McCarthy MI, Abecasis GR, Cardon LR, Goldstein DB, Little J, Ioannidis JP, Hirschhorn JN. 2008. Genome-wide association studies for complex traits: consensus, uncertainty and challenges. Nat Rev Genet 9(5):356-69.
Morris MC, Beckett LA, Scherr PA, Hebert LE, Bennett DA, Field TS, Evans DA. 1998. Vitamin E and vitamin C supplement use and risk of incident Alzheimer disease. Alzheimer Dis Assoc Disord 12(3):121-6.
Morris RW, Kaplan NL. 2002. On the advantage of haplotype analysis in the presence of multiple disease susceptibility alleles. Genet Epidemiol 23(3):221-33.
Moskvina V, Schmidt KM, Vedernikov A, Owen MJ, Craddock N, Holmans P, O'Donovan MC. 2012. Permutation-based approaches do not adequately allow for linkage disequilibrium in gene-wide multi-locus association analysis. Eur J Hum Genet 20(8):890-6.
Motsinger-Reif AA. 2008. The effect of alternative permutation testing strategies on the performance of multifactor dimensionality reduction. BMC Res Notes 1:139.
Neale BM, Sham PC. 2004. The future of association studies: gene-based analysis and replication. Am J Hum Genet 75(3):353-62.
Neuhauser M, Bretz F. 2005. Adaptive designs based on the truncated product method. BMC Med Res Methodol 5:30.
North BV, Curtis D, Sham PC. 2002. A note on the calculation of empirical P values from Monte Carlo procedures. Am J Hum Genet 71(2):439-41.
Ovcharenko I, Loots GG, Nobrega MA, Hardison RC, Miller W, Stubbs L. 2005. Evolution and functional classification of vertebrate gene deserts. Genome Res 15(1):137-45.
Patil N, Berno AJ, Hinds DA, Barrett WA, Doshi JM, Hacker CR, Kautzer CR, Lee DH, Marjoribanks C, McDonough DP and others. 2001. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. Science 294(5547):1719-23.
Pattaro C, Ruczinski I, Fallin DM, Parmigiani G. 2008. Haplotype block partitioning as a tool for dimensionality reduction in SNP association studies. BMC Genomics 9:405.
Perry RT, Collins JS, Wiener H, Acton R, Go RC. 2001. The role of TNF and its receptors in Alzheimer's disease. Neurobiol Aging 22(6):873-83.
Phipson B, Smyth GK. 2010. Permutation P-values should never be zero: calculating exact P-values when permutations are randomly drawn. Stat Appl Genet Mol Biol 9(1):Article39.
Potter DM. 2006. Omnibus permutation tests of the association of an ensemble of genetic markers with disease in case-control studies. Genet Epidemiol 30(5):438-46.
Potter DM, Griffiths DJ. 2006. Omnibus permutation tests of the overall null hypothesis in datasets with many covariates. J Biopharm Stat 16(3):327-41.
Price AL, Zaitlen NA, Reich D, Patterson N. 2010. New approaches to population stratification in genome-wide association studies. Nat Rev Genet 11(7):459-63.
Pruim RJ, Welch RP, Sanna S, Teslovich TM, Chines PS, Gliedt TP, Boehnke M, Abecasis GR, Willer CJ. 2010. LocusZoom: regional visualization of genome-wide association scan results. Bioinformatics 26(18):2336-7.
Purcell S, Daly MJ, Sham PC. 2007a. WHAP: haplotype-based association analysis. Bioinformatics 23(2):255-6.

Purcell S, Neale B, Todd-Brown K, Thomas L, Ferreira MA, Bender D, Maller J, Sklar P, de Bakker PI, Daly MJ and others. 2007b. PLINK: a tool set for whole-genome association and population-based linkage analyses. Am J Hum Genet 81(3):559-75.
Qin ZS, Niu T, Liu JS. 2002. Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. Am J Hum Genet 71(5):1242-7.
R Development Core Team. 2012. R: A language and environment for statistical computing: R Foundation for Statistical Computing.
Raeymaekers P, Timmerman V, Nelis E, De Jonghe P, Hoogendijk JE, Baas F, Barker DF, Martin JJ, De Visser M, Bolhuis PA and others. 1991. Duplication in chromosome 17p11.2 in Charcot-Marie-Tooth neuropathy type 1a (CMT 1a). The HMSN Collaborative Research Group. Neuromuscul Disord 1(2):93-7.
Ramakrishnan B, Boeggeman E, Qasba PK. 2012. Binding of N-acetylglucosamine (GlcNAc) beta1-6-branched oligosaccharide acceptors to beta4-galactosyltransferase I reveals a new ligand binding mode. J Biol Chem 287(34):28666-74.
Roses AD, Lutz MW, Amrine-Madsen H, Saunders AM, Crenshaw DG, Sundseth SS, Huentelman MJ, Welsh-Bohmer KA, Reiman EM. 2010. A TOMM40 variable-length polymorphism predicts the age of late-onset Alzheimer's disease. Pharmacogenomics J 10(5):375-84.
Schwartz R, Halldorsson BV, Bafna V, Clark AG, Istrail S. 2003. Robustness of inference of haplotype block structure. J Comput Biol 10(1):13-9.
Sertie AL, Sossi V, Camargo AA, Zatz M, Brahe C, Passos-Bueno MR. 2000. Collagen XVIII, containing an endogenous inhibitor of angiogenesis and tumor growth, plays a critical role in the maintenance of retinal structure and in neural tube closure (Knobloch syndrome). Hum Mol Genet 9(13):2051-8.
Sham PC, Rijsdijk FV, Knight J, Makoff A, North B, Curtis D. 2004. Haplotype association analysis of discrete and continuous traits using mixture of regression models. Behav Genet 34(2):207-14.
Simes RJ. 1986. An improved Bonferroni procedure for multiple tests of significance. Biometrika 73(3):751-754.
Skol AD, Scott LJ, Abecasis GR, Boehnke M. 2006. Joint analysis is more efficient than replication-based analysis for two-stage genome-wide association studies. Nat Genet 38(2):209-13.
Slooter AJ, Cruts M, Kalmijn S, Hofman A, Breteler MM, Van Broeckhoven C, van Duijn CM. 1998. Risk estimates of dementia by apolipoprotein E genotypes from a population-based incidence study: the Rotterdam Study. Arch Neurol 55(7):964-8.
Staples G. 2006. TORQUE resource manager. Proceedings of the 2006 ACM/IEEE conference on Supercomputing. Tampa, Florida: ACM. p 8.
Stein LD. 2004. Human genome: end of the beginning. Nature 431(7011):915-6.
Steiss V, Letschert T, Schafer H, Pahl R. 2012. PERMORY-MPI: a program for high-speed parallel permutation testing in genome-wide association studies. Bioinformatics 28(8):1168-9.
Su Z, Marchini J, Donnelly P. 2011. HAPGEN2: simulation of multiple disease SNPs. Bioinformatics 27(16):2304-5.
Swardfager W, Lanctot K, Rothenburg L, Wong A, Cappell J, Herrmann N. 2010. A meta-analysis of cytokines in Alzheimer's disease. Biol Psychiatry 68(10):930-41.

Sweetman D, Munsterberg A. 2006. The vertebrate spalt genes in development and disease. Dev Biol 293(2):285-93.
Takeuchi T, Adachi Y, Nagayama T. 2011. Expression of a secretory protein C1qTNF6, a C1qTNF family member, in hepatocellular carcinoma. Anal Cell Pathol (Amst) 34(3):113-21.
Tang CS, Ferreira MA. 2012. A gene-based test of association using canonical correlation analysis. Bioinformatics 28(6):845-50.
Thain D, Tannenbaum T, Livny M. 2005. Distributed computing in practice: the Condor experience: Research Articles. Concurr. Comput. : Pract. Exper. 17(2-4):323-356.
Thorisson GA, Smith AV, Krishnan L, Stein LD. 2005. The International HapMap Project Web site. Genome Res 15(11):1592-3.
Timpson NJ, Forouhi NG, Brion MJ, Harbord RM, Cook DG, Johnson P, McConnachie A, Morris RW, Rodriguez S, Luan J and others. 2010. Genetic variation at the SLC23A1 locus is associated with circulating concentrations of L-ascorbic acid (vitamin C): evidence from 5 independent studies with >15,000 participants. Am J Clin Nutr 92(2):375-82.
Tippett LHC. 1952. The methods of statistics: Williams and Norgate.
Tomasi ML, Li TW, Li M, Mato JM, Lu SC. 2012. Inhibition of human methionine adenosyltransferase 1A transcription by coding region methylation. J Cell Physiol 227(4):1583-91.
Varki A, Freeze HH. 2009. Glycans in Acquired Human Diseases. In: Varki A, Cummings RD, Esko JD, Freeze HH, Stanley P, Bertozzi CR, Hart GW, Etzler ME, editors. Essentials of Glycobiology. 2nd ed. Cold Spring Harbor (NY).
Velez JI, Chandrasekharappa SC, Henao E, Martinez AF, Harper U, Jones M, Solomon BD, Lopez L, Garcia G, Aguirre-Acevedo DC and others. 2012. Pooling/bootstrap-based GWAS (pbGWAS) identifies new loci modifying the age of onset in PSEN1 p.Glu280Ala Alzheimer's disease. Mol Psychiatry.
Verghese PB, Castellano JM, Holtzman DM. 2011. Apolipoprotein E in Alzheimer's disease and other neurological disorders. Lancet Neurol 10(3):241-52.
Veyrieras JB, Kudaravalli S, Kim SY, Dermitzakis ET, Gilad Y, Stephens M, Pritchard JK. 2008. High-resolution mapping of expression-QTLs yields insight into human gene regulation. PLoS Genet 4(10):e1000214.
Wang N, Akey JM, Zhang K, Chakraborty R, Jin L. 2002. Distribution of recombination crossovers and the origin of haplotype blocks: the interplay of population history, recombination, and mutation. Am J Hum Genet 71(5):1227-34.
Wigginton JE, Cutler DJ, Abecasis GR. 2005. A note on exact tests of Hardy-Weinberg equilibrium. Am J Hum Genet 76(5):887-93.
Woo RS, Lee JH, Yu HN, Song DY, Baik TK. 2010. Expression of ErbB4 in the apoptotic neurons of Alzheimer's disease brain. Anat Cell Biol 43(4):332-9.
Woo RS, Lee JH, Yu HN, Song DY, Baik TK. 2011. Expression of ErbB4 in the neurons of Alzheimer's disease brain and APP/PS1 mice, a model of Alzheimer's disease. Anat Cell Biol 44(2):116-27.
Yu CE, Seltman H, Peskind ER, Galloway N, Zhou PX, Rosenthal E, Wijsman EM, Tsuang DW, Devlin B, Schellenberg GD. 2007. Comprehensive analysis of APOE and selected proximate markers for late-onset Alzheimer's disease: patterns of linkage disequilibrium and disease/marker association. Genomics 89(6):655-65.

Zandi PP, Anthony JC, Khachaturian AS, Stone SV, Gustafson D, Tschanz JT, Norton MC, Welsh-Bohmer KA, Breitner JC. 2004. Reduced risk of Alzheimer disease in users of antioxidant vitamin supplements: the Cache County Study. Arch Neurol 61(1):82-8.
Zaykin DV, Zhivotovsky LA, Westfall PH, Weir BS. 2002. Truncated product method for combining P-values. Genet Epidemiol 22(2):170-85.
Zhang K, Deng M, Chen T, Waterman MS, Sun F. 2002. A dynamic programming algorithm for haplotype block partitioning. Proc Natl Acad Sci U S A 99(11):7335-9.
Zhang K, Qin Z, Chen T, Liu JS, Waterman MS, Sun F. 2005. HapBlock: haplotype block partitioning and tag SNP selection software using a set of dynamic programming algorithms. Bioinformatics 21(1):131-4.
Zhang K, Qin ZS, Liu JS, Chen T, Waterman MS, Sun F. 2004. Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies. Genome Res 14(5):908-16.
Zheng X, Levine D, Shen J, Gogarten SM, Laurie C, Weir BS. 2012. A High-performance Computing Toolset for Relatedness and Principal Component Analysis of SNP Data. Bioinformatics.
Zhou C, Zang D, Jin Y, Wu H, Liu Z, Du J, Zhang J. 2011. Mutation in ribosomal protein L21 underlies hereditary hypotrichosis simplex. Hum Mutat 32(7):710-4.