Exploration of SNP-set Interactions in Genome-Wide Association Studies

by

Shomoita Alam

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

in

Epidemiology

Department of Public Health Sciences

University of Alberta

Abstract

Advance in biotechnologies has enabled genome-wide association studies (GWAS) that scan the entire human genome for understanding genetic contributions to the risk of a certain disease as well as to variation in treatment efficacy and side effects. In GWAS, the association between each single-nucleotide polymorphism (SNP) and a phenotype is assessed statistically, typically analyzing one single SNP at a time, ignoring potential SNP-SNP interactions. Such individual-SNP analysis approaches have extracted small fractions of expected genetic contributions to disease risks: this has been recognized as "the missing heritability problem" of GWAS. Biologically, it is highly unlikely that a single SNP alone would determine disease risk, especially for complex chronic diseases. We therefore tested whether biological interactions among multiple SNPs determine disease risks and whether it can explain the missing heritability problem. The methodologies proposed in this work take into account the interaction between selected SNP-sets using two methods: (1) method based on logic regression that incorporates two specific forms of interaction; and (2) method based on SNP-pair analysis which is an exploration of genotypes that are only observed in cases with a sufficient frequency and with no control having the same specific genotypes. Both methods could identify many previously-found and novel susceptibility genes for the datasets we tested on, although validation studies are required to avoid spurious findings. While our results do not provide a satisfactory solution to the "missing heritability" problem, they show

the importance of considering SNP interactions and their exploration in considering genetic contributions of disease etiology, prevention and treatment.

Preface

This thesis is an original work by Shomoita Alam. No part of this thesis has been previously published.

Contents

List of Tables

List of Figures

List of Abbreviations

AS Ankylosing Spondylitis

CD Crohn's Disease

GWAS Genome-Wide Association Studies

IBD Inflammatory Bowel Disease

LD Linkage Disequilibrium

MS Multiple Sclerosis

OR Odds Ratio

RA Rheumatoid Arthritis

SLE Systemic Lupus Erythematosus

SNP Single-Nucleotide Polymorphism

T1D Type 1 Diabetes

T2D Type 2 Diabetes

UC Ulcerative Colitis

WTCCC Wellcome Trust Case Control Consortium

Chapter 1

Introduction

1.1 Genome-Wide Association Studies

A genome-wide association study (GWAS) is an approach that involves "markers" across the whole genomes in a population to find genetic variations associated with a particular disease or trait. Researchers usually use two groups of participants to carry out a GWAS: people with the disease (cases) being studied and people from the same population without the disease (controls). From each participant a DNA sample is obtained, usually by drawing blood, rubbing a cotton swab along the inside of the mouth to harvest cells or squish and spit. Those samples are then processed to extract DNA and placed on tiny chips and scanned by an automated reader. Single nucleotide polymorphisms (SNPs) are the markers measured by these chips. The genetic variations of SNP genotypes are said to be associated with the disease, if certain patterns of SNP genotypes are found to be statistically significantly more frequent in people with

the disease compared to people without disease[1].

GWAS was enabled by the entire human genome sequencing which has allowed for subsequent identification and cataloguing of human genetic variation. As a result, a map of human genome including that of more than three million SNPs has been generated[2]. It is considered that a large portion of the human variation can be surveyed by genotyping a limited number of SNPs (tag SNPs) that are representative of all human haplotypes. SNP genotypes tend to be correlated with genotypes of other SNPs of nearby region; this means that genotyping only a few carefully chosen SNPs will provide enough information about the remainder of the common SNPs in that region. To evaluate human genetic variation genome-wide SNP arrays have been constructed by simultaneously genotyping for millions of SNPs[3].

Discovery of genetic markers that are associated with diseases and understanding the mechanism of how these genetic variations contribute to disease etiologies will gain insights into the prevention, diagnosis and treatment of human disease. GWAS are considered to be particularly useful for studying genetic contributions to common and complex diseases, such as cardiovascular disease, cancer, obesity, diabetes, psychiatric illnesses and inflammatory diseases as they are caused by combination of multiple genetic and environmental factors[4].

1.2 Advances in GWAS

In the past decade GWAS has become a major framework, often conducted as multi-country collaborative projects and have seen many scientific and biological discoveries. These studies were aimed at detecting associations between common SNPs and common diseases such as heart disease, diabetes, auto-immune diseases, and psychiatric disorders[5]. Currently, over 1400 GWAS have uncovered more than 7,000 associations between SNPs and more than 700 traits/diseases[6]. While case-control designs are the most common, other study designs have been employed for GWAS, including cohort studies, clinical trials and trio designs[7,8].

The initial results of GWAS were reported in 2005[9] and 2006[10] in Science. The GWAS conducted by the Wellcome Trust Case Control Consortium (WTCCC) that was published in Nature[11] in 2007 is considered a breakthrough study at the inception of GWAS era. The reason for this is that the WTCCC study was the first large, well-designed GWAS for complex diseases to employ a SNP chip that had good coverage of the whole genome.

A vast majority of new loci have been identified after 2007, having association with several diseases or complex traits. The number of loci identified per complex trait varies substantially, from a handful for psychiatric diseases to a hundred or more for inflammatory bowel disease (IBD) (including Crohn's disease (CD)[12] and ulcerative colitis (UC)[13]) and stature. It has been observed that the number of discovered variants is strongly correlated with study sample size, i.e., with an increasing sample size the

number of discovered variants will generally increase[5].

GWAS on auto-immune diseases such as ankylosing spondylitis (AS), rheumatoid arthritis (RA), systemic lupus erythematosus (SLE), type 1 diabetes (T1D), multiple sclerosis (MS), CD and UC have identified a large number of genes associated with these diseases. According to the paper by Visscher *et al.*[5] who reviewed GWA studies from 2007 to 2012, 19 loci were known prior to 2007 and more than 277 have been discovered from 2007 onward across these diseases. GWAS have been successful not only in terms of discovering the number of loci. It has also discovered shared loci between diseases highlighting a number of pathways of their mechanistic continuum and uncovering potential theraputic targets. Some of these pathways were previously not suspected to be important for these diseases. For example, while CD and UC display distinct clinical feature, it has been suggested through GWAS findings that these two diseases share certain pathways. There are also strong overlaps between genes involved in CD, UC, AS and psoriasis, suggesting shared aetiopathogenic mechanisms in these conditions[14].

In terms of metabolic diseases such as type 2 diabetes (T2D), GWAS have published more than 50 genome-wide-significant hits involving individuals of European descent, East Asians, South Asians, Hispanic and African Americans; not many of them were known before the GWAS era. Prior to the GWAS era, the only compelling association signal for fasting glucose levels which is a quantitative trait, was known at GCK (glucokinase). GWAS in European samples have expanded that number to 1632. These variants explain around 10% of the inherited variation in fasting glucose levels. Al-

though the GWAS approach was designed for the detection of associations between DNA markers and disease, as a by-product such studies have generated new biology and scientific discoveries such as the discovery of genes affecting genetic recombination and their correlation with natural selection and new insight in human population structure and evolution[5].

1.3 Missing Heritability and GWAS

The GWAS approach represents an important step compared to the candidate gene studies and family-based linkage studies. The underlying rationale for GWAS is the 'common disease, common variant' hypothesis, that states that a disease that is relatively common results from the joint action of multiple common genetic variants[15]. The commercial SNP chips intend to capture most, although certainly not all of the common variations in the genome. The candidate gene approach has been largely explored to study complex diseases. This approach focuses on genes that are selected based on a priori hypothesis about their etiological role in disease and usually conducted in a population-based study. Therefore, candidate gene studies take the advantage of both the increased statistical efficiency of association analysis and the biological understanding of the phenotype, tissues, genes and proteins that are likely to be involved in the disease. However, this approach has many criticisms because of non-replication of results and its limited ability to include all possible causative genes and polymorphisms[16]. Family-based linkage studies are designed based on related individuals and has led to

the discovery of many genes for Mendelian diseases and traits. This design, as opposed to population based studies avoids problems of population heterogeneity and stratification. However, features such as sensitivity to genotyping error make family-based linkage studies less attractive to their population-based counterparts, as false inferences can be drawn because the test distribution depends on the assumption that parental genotypes are correct[17,18,19].

GWAS is conducted often in multi-country collaborative projects, aiming to identify regions of human genome whose variations across subjects are associated with risk of various diseases. However, the known degree of genetic contribution to the risk of developing a particular disease, known as "heritability" has been explained rather poorly in GWAS findings. Thus for only a small fraction of the known degree of the "heritability" are attributed to hereditary components discovered by GWAS, leading to a phenomenon called *missing heritability*. Specifically, GWAS is an epidemiological case-control study, where the disease risk may be modified by SNP genotypes or environmental exposures of interest. Typically in case-control studies the measures of association used is odds ratio (OR) and this is estimated in GWAS. Departure from 1.0 in the ORs indicates that corresponding genotypes of SNPs or the region in the genome tagged by them are associated with the disease risk[20]. In the vast majority of GWAS conducted to-date, however, most common variants individually or in combination confer relatively small increments in ORs, usually OR < 1.5 and explain only a small proportion of heritability known from family based studies[21].

The potential causes of missing heritability have been widely debated and many ex-

planations have been suggested. In early GWAS, limitations in the study designs such as imprecise phenotyping, questionable control groups may have reduced the estimates of the effect size. Other explanations of missing heritability include rarer variants possibly with larger effects that are poorly detected by available chips, structural variants poorly captured by existing arrays, low power to detect gene–gene interactions and inadequate accounting for environment exposures and their sharing relatives[22,21].

1.4 Rationale and Objectives

Determining which variants are associated with a given phenotype out of the millions of variants in an individual's genome is a massive task, especially if the functional consequences of the causal alleles are poorly known. It is critical to determine effective approaches for combining the functional credibility and statistical support in the evaluation of such variants. Majority of the GWAS studies have tended to focus almost exclusively on statistical evidence without much consideration of biological plausibility. The challenge is to strategize a way to group the rare variants, separating them out of the million variants and analysing them properly[21].

The "missing heritability" problem may be tackled as elaborated in Yasui[20], using statistical reasoning by incorporating the concepts of redundancy and concurrence in the functions of SNPs, especially within the set of genes on the same pathway in association with a phenotype. Interaction of redundancy occurs when SNP is strongly associated with the increased disease risk in the absence of another SNP, but the presence of that

other SNP masks the SNP-disease association. Interaction of concurrence occurs when SNP is strongly associated with the disease risk only in the presence of another SNP. These specific forms of interactions have been largely unexplored in GWAS where the standard data analysis examines each single SNP one at a time.

The underlying biology of complex diseases studied in GWAS, however, is expected to involve multiple SNPs and non-genetic factors. Individual SNPs cannot represent accurately the extent and mechanism of the underlying disease biology when multiple causal factors are involved. Therefore, to make proper statistical evaluation of the genetic contributions with the phenotype, these specific forms of interactions should be fully explored, assuming that heritability information itself is not missing in GWAS data. The broad objective of this thesis is to explore interactions among many genetic variants with the intention to explain missing heritability.

In this thesis, we will discuss several approaches to explore interactions among the genetic variants in GWAS settings. In Chapter 2, we will discuss how incorporating the concepts of interaction is important compared to the single SNP analysis and explore some existing methodologies of implementing them. In Chapter 3, we will explain the methods we used to explore SNP-set interactions and establish the results of those methods in different GWAS data. Chapter 4 will include the discussion and conclusion of the methodologies explored in this thesis.

Chapter 2

Importance of Interaction in GWAS

2.1 Approaches for Analysis of GWAS Data

2.1.1 Single SNP Analysis

The standard approach for analyzing GWAS in the discovery phase involves individual SNP analysis. In this method the phenotype is regressed onto each individual SNP or a test is performed for phenotype association (e.g., Chi square test or Fisher's Exact Test) for each SNP. The SNPs are ranked based on their individual p-values and the p-values that are less than a threshold, which is set beforehand, are pushed forward for validation. The threshold can be based on controlling the Type I error probability in multiple testing of many SNPs by Bonferroni's or other correction method. However,

this threshold for genome wide significance can be very difficult to attain due to the large number of considered hypotheses: for example, a GWAS examining the effects of 500,000 SNPs, each test is conducted at the threshold 10^{-7} level, which is very stringent. Now, the newest chip contains millions of SNPs and genetic markers.

Analyzing individual SNP has certain limitations. First, the sample size of the study is required to be very large (providing high power) so that statistical significance of truly-positive SNPs will be retained after a multiple-testing correction of Type I error probability. It increases the investigation cost and leaves many borderline significant SNPs inconclusive. Second, individual-SNP analysis is often limited by poor reproducibility as many of the highly-ranked SNPs cannot be validated due to false positives. In particular, individual SNPs that are genotyped on GWAS platforms often show only small or modest effects. The explanation for this may be that the true causal SNP is rarely genotyped, but there are typed SNPs which are in linkage disequilibrium (LD) with the causal SNP. As the typed SNPs in LD serve as an imperfect surrogate (tag) for the causal SNP, such individual SNPs will only show small effects. Finally, epistasis interactions between SNPs are ignored when only the marginal effect of individual SNP analysis is considered[20]. Interaction between SNPs can contribute to disease susceptibility such that interaction between SNPs can have a larger effect whereas single SNPs may show little individual effect. There is considerable biological interaction present in the genomic pathway of a disease which is not possible to detect with single SNP analysis[23].

2.1.2 Importance of Incorporating Interaction

Biologists and quantitative geneticists consider that a set of genes (SNPs) is responsible for the association with a specific phenotype instead of a single gene (SNP). To understand the extent and mechanism of the underlying biology of the complex diseases studied in GWAS it is required to incorporate the idea of interaction among multiple SNPs. We propose to incorporate interaction in the analysis of GWAS data to properly evaluate the association between SNPs and phenotype statistically. Two important concepts that are key to realize how the SNPs are interacting is *interaction of redundancy* and *interaction of concurrence*.

Two or more SNPs may be associated with disease risk such that either is sufficient to modify disease risk, but neither is necessary; this is an example of redundancy. For example, for a certain biological function, interaction of redundancy of two SNPs A and B is expressed mathematically as $A \cup B$ and it means that having either A or B is sufficient and the presence of both does not result in the sum of the two effects. The concept of concurrence can be explained with a scenario if modification of disease risk requires multiple factors to take place concurrently. For example, interaction of concurrence of two SNPs A and B, is expressed mathematically as $A \cap B$ and it means that both A and B are required to activate a certain biological function[20].

2.2 Methods for Detection of SNP Interaction

There are several challenges to overcome to be able to successfully detect loci associated with complex diseases in GWAS. One of the challenges is to incorporate the specific forms of interaction, i.e., redundancy and concurrence. The traditional statistical methods may be insufficient to capture the high dimensional interactions among the GWAS data. Another challenge is that the nature of SNP data sets makes SNP interaction identification a combinatorial search problem with huge amounts of SNP information. The searching procedure becomes computationally intensive due to the the massive number of possible combinations of SNP interaction. Various methods have been proposed to either resolve any of the two challenges or finding a balance between them[24]. Two existing methods will be described in the following sections: (1) logic regression[25], a method to incorporate two specific forms of interaction and to find a good fitting model from a large search space of possible models; and (2) two stage testing procedure[26] that filters the SNPs based on marginal association and tests for interaction only among the filtered SNPs so that the search space for the test of interaction is smaller compared to search among all the SNPs.

2.2.1 Logic Regression

Logic regression[25] is a generalized linear model designed to model an outcome (e.g., phenotype in case-control study) with various *intersections* and/or *unions* of potential binary predictors that are associated with a phenotype, such as SNP genotypes (i.e.,

indicators of the minor-allele homozygous, indicators of the heterozygous and indicators of the major-allele homozygous) as potential predictors. Combinations of SNP intersections and unions can be expressed mathematically as Boolean combinations, such as $(X_1 \cap X_2) \cup X_3^c$, where $\cap$, $\cup$ and c represents intersection (AND), union (OR), and complement (NOT), and X's are indicators of SNP genotypes. The model can be described as follows:

$$g(E(Y)) = \beta_0 + \sum_{i=1}^{n} \beta_i L_i \tag{2.1}$$

where g is a link function, Y is the response, L_i represents a Boolean combination of the binaries, also called logic trees, and β_i's denote the regression parameters. In case-control GWASs, g is usually given by $logit(x) = \log(\frac{x}{1-x})$, where $x \in (0, 1)$.

To estimate the regression coefficients, βs, the method tries to find the Boolean expressions that minimize an optimization function (e.g., deviance function for logit link) which indicates the "fit" of the model. Since the number of possible logic models we can construct for a given set of SNPs is very large, a maximum number, n, of Boolean predictors and a maximum "tree size" of Boolean predictors are prespecified. Logic regression uses a Simulated Annealing algorithm to find the "best" model from the large search space of possible models. To avoid over-fitting, logic regression could use a K-fold cross-validation to determine the maximum tree size to search instead of prespecifying it.

Logic regression is a method for incorporating the concepts of redundancy and concurrence to analyze GWAS data as it uses Boolean expressions as predictors. This

method has been successfully applied to SNP data analysis with selected candidate genes to explain the disease genetics of highly heritable diseases[27,28,29,30]. More recently, our team has applied it to GWAS with a limited tree-size and a limited number of SNPs within genes to form the logics.

2.2.2 Two Stage Testing Procedure

Advanced sequencing methods and high-throughput technologies have made it possible to characterize millions of sequence variations on large numbers of study participants. However, one of the challenges that investigators face while identifying a small number of these genetic features that are associated with a disease trait is that large number of unrelated genetic features have to be examined together with the small number of biologically relevant features. To tackle this challenge, two-stage multiple testing procedures have been proposed where the idea is to filter out the majority of the irrelevant genetic variants initially and only test for interaction among the promising variants[31,32,26].

Dai *et al.*[26] has explained and proved theoretically how two stage procedure with independent filtering method can be justified to detect gene-gene and gene-environment interactions. One type of filtering statistic is considered to be the marginal association of the genetic variant[31]. The statistic used in the filtering stage is shown to be *asymptotically independent* of the statistic in the testing stage where the interaction is tested under the null hypothesis of no interaction, so that multiple testing correction is only needed for the tests that actually pass the filtering, thereby potentially improv-

ing power. The authors formulated their paper in the context of gene-environment interactions only and the theorem is as follows.

Let Y be an outcome variable in a generalized linear model with a canonical link function g, X be the genetic variable, Z be the environmental variable and W be the additional covariates. Consider two nested generalized linear models:

$$g(E(Y|X,W)) = \beta_0 + \beta_1 X + \beta_2 W, \tag{2.2}$$

$$g(E(Y|X,Z,W)) = \gamma_0 + \gamma_1 X + \gamma_2 Z + \gamma_3 XZ + \gamma_4 W. \tag{2.3}$$

Then the maximum likelihood estimator $\hat{\beta}_1$ and $\hat{\gamma}_3$ are asymptotically independent.

The key idea is that dimensionality of multiple testing in genomics can be reduced by screening features to be tested with an independent statistic in the same dataset, thereby mitigating the multiple-testing problem and increasing power to detect effects. This implies that the noise is reduced and it allows for relevant signals to be more easily detected. The application of two stage procedure can be used as a data adaptive tool, as opposed to candidate genes from prior studies, for discovering novel genes that affect disease risk. This method is likely to gain importance as the high-throughput technologies continue to yield exponentially increasing amount of information per sample in every research conducted.

The theoretical properties detailed by Dai *et al.*[26] apply not only to search for gene-environment interaction, but also for gene-gene interaction, since both "gene" and "environment" features are treated analogously as discrete or continuous variables in models designed to identify associations with a disease trait in constructing these

hypothesis tests[33]. Therefore, filtering SNPs by marginal association and testing for SNP-SNP interaction among only the chosen set of SNPs would mitigate the dimensionality problem.

Chapter 3

Analysis and Results

Our hypothesis is that interaction is key to understand the disease genetics in GWAS data. To explore interaction among the SNP sets in GWAS data, we propose three methodologies described in the following sections.

3.1 Logic Regression across Whole Genome

3.1.1 Analysis

SNP-SNP interactions at the *gene-level* have been explored using logic regression by Dinu *et al.*[27] in the CD GWAS data of the WTCCC[11]. In this study 195 genes have been discovered to have strong evidence of CD-association including some previously identified genes and novel susceptibility genes. However, we identified in a previous work that 72 of these top genes are results of genotyping error by WTCCC. Genotyping errors lead to false positive association[34]. The technology and calling algorithms of WTCCC

have resulted in relatively high occurrences of suspected errors in genotype calling by inspecting the intensity plots of genotype calling manually.

Proposal to Study SNP-SNP Interaction

Genotyping error can be checked visually by plotting the mean adjusted intensities for each allele against each other where each color represents a different genotype in the cluster plots. The x and y axes on the plots denote the intensity measurements for the two alleles of the SNP. Each point represents the measurement for a single individual. For both cases and controls, SNP genotype cluster intensity plots needs to be generated. SNPs whose plots would indicate potential genotyping errors would be excluded.

For a high quality marker, clusters of different colors (genotypes) should be separated from each other, indicating a high confidence in genotype calling. Figure 3.1 shows an example of correct genotype calling for SNP rs6752107 as clear separation of the genotype clusters can be observed which indicates a high quality marker. For a low quality marker, clusters of different colors (genotypes) might not have clear boundaries and overlap with each other indicating genotyping error. An example of such genotyping error plot can be depicted in Figure 3.2 for SNP rs2314349. The intensity plots of cases and controls show obvious overlaps of GG and GT indicating possible error in genotype calls, which could result in false positive association.

We performed logic regression analysis as described in Subsection 2.2.1 with all the top 123 genes (4066 SNPs) found from the *gene-level analysis* after removing the 72 genes with genotyping error from the WTCCC CD data. This logic regression consid-

ered all the SNPs of the 123 genes together so that the SNPs from different chromosomes can interact with each other, as opposed to the gene-level analysis where SNPs could only interact with the SNPs within a gene. We used a 10-fold cross validation technique to select the optimum number of leaves and trees for the logic regression. For the purpose of selecting the optimum number of trees and leaves, we explored a range of the number of trees from 1 to 9 and a range of the number of leaves from 1 to 20. The model for 10-fold cross validation was run independently 30 times varying the seed of the random number generator at the beginning of the stochastic search of logic regression. The logic behind varying the seeds is to be able to search more broadly into the solution space and reduce the probability of converging to a local optimum. After we selected the optimum number of leaves and trees we ran the logic regression model with 30 different randomly generated starting points (seeds) to get the best fit to the logic regression model of the given size selecting the model having the lowest deviance.

Checking genotyping error manually for all the SNPs of the 123 top genes for CD would be computationally expensive. To make sure that the current model does not have any SNP with genotyping error, we checked genotyping error for the SNPs appearing in the logic regression model and repeated the following steps to get a final model with no SNPs with genotyping error.

1. Genotyping error was checked for the current model of the size of the best fitted model.

2. Manual examination of genotyping error as described above is done and erroneous

SNPs were removed.

3. Logic regression was run again with the given size to get a new set of interacting SNPs.

4. The latest best model was selected based on the minimum deviance criteria among 30 iterations with different seeds.

5. Step 1-4 was repeated until all the SNPs of the best fitted model of the given size contain no SNP with genotyping error.

3.1.2 Results

The resulting optimum number for the trees and leaves combination was 9 trees and 17 leaves, which were selected based on the minimum of the minimum prediction score produced from the 10-fold cross validation models generated using 30 different seeds. The model of the given size was checked iteratively for genotyping error as explained above and the final logic regression model was obtained after nine iterations of genotyping error check. We removed 37 erroneous SNPs. This model was obtained analyzing 4029 SNPs for 1748 CD cases and 2936 controls.

We found the 17 SNPs showed in Table 3.1 in the best logic regression model after analyzing all the SNPs of the top 123 genes that had no indication of possible genotyping error. The table provides summary of each SNP along with the percentage of those SNPs in cases and controls. The logic trees formed with these 17 SNPs are displayed in Table 3.2 along with the associated odds ratios. From Table 3.2, we can observe

that two SNPs, namely rs17221417 (NOD2 gene) and rs17234657 (PTGER4 gene) appeared independently to be associated with the phenotype (CD) in the interaction model with odds ratios 0.59 and 0.67 (OR=1.49 for the absence of this SNP), respectively. Examples of interaction of redundancy appeared in few of the logic trees, for example, "rs9858542 or rs6752107" shows that either SNP rs9858542 (BSN gene) or SNP rs6752107 (ATG16L1 gene) is sufficient for the disease to occur with associated odds ratio 1.75. Interaction of concurrence can be exemplified by the logic tree "(rs7515029 and (not rs7807268))", which implies that presence of SNP rs7515029 (C1orf141 gene) and absence of SNP rs7807268 (C7orf33 gene) concurrently is necessary for the disease risk to be elevated and the associated odds ratio for this combination was estimated to be 1.51. The odds ratios for 8 of the logic trees showing interactions of SNPs are ranging from 1.49 to 1.75 and one logic tree, "((not rs888775) or (not rs6674713))", appears to have a high odds ratio (27.24). It can be observed from Table 3.1, that SNP rs888775 (WWC1 gene) and SNP rs6674713 (PTCHD2 gene) have prevalence of 100% (or close to it) in controls.

We have found 11 genes (12 SNPs), namely ZNF365, BSN, ATG16L1, MST150, PTPN2, NKD1, IL23R, C1orf141, NOD2, PTGER4 (appearing twice with two different SNPs) and NKX2-3 to be overlapping with the genes/SNPs/chromosomal locations showing strong evidence of association by the single-SNP analysis of WTCCC data. One SNP (rs7807268, Chr 7q36.1) showing moderate association in the WTCCC paper was also found in our analysis. We have found four genes (SNPs) that were not reported in WTCCC single SNP analysis, namely, CERKL, NRXN1, WWC1 and

PTCHD2. Thirteen (76%) genomic regions (7 genes, 2 chromosomal locations, 1 SNP and 3 SNPs/genes) that were previously identified by the WTCCC single-SNP study were included in the 17 genomic regions that we discovered. Moreover, fourteen (82%) genomic region (5 chromosomal location and 9 gene) that we found overlaps with the meta-analysis of single-SNP studies that involved over 22,000 cases and 29,000 controls. Three genes, namely CERKL, C7orf33 and WWC1, were not identified in the single-SNP analyses that were indicated in this meta-analysis. Table 3.3 compares our results with the results of WTCCC single-SNP analysis and the meta-analysis based on overlapping SNPs, genes and chromosomal locations for CD.

To check if the interaction model is performing better than a model that can be formed with the main effects of 17 SNPs, we used Clarke's non-nested model test[35,36] to compare the two models. Two models are non-nested if one model cannot be reduced to the other model by imposing a set of linear restrictions on the parameter vector or they are non-nested in terms of their functional forms. The interaction model and the 17 SNP main effects model will be non-nested as we cannot impose a set of linear restriction on the parameter vector of one model to get another model. Usual tests to discriminate between models such as the likelihood ratio test cannot compare two models if they are not nested.

Our interaction model and the 17 SNP main effects model are partially non-nested. Thus, we performed Clarke's non-nested model test to discriminate between the two models. The reason behind using Clarke's non-nested model test is that it is a distribution free test and it can differentiate between non-nested and partially non-nested

models. This test applies a modified paired sign test to the differences in the individual log-likelihoods from two nonnested models and determines whether or not the median log-likelihood ratio is statistically significantly different from zero. The test statistic is simply the number of positive differences in the individual log-likelihoods, which has a Binomial distribution with parameter n (number of observations) and $\theta = 0.5$. An average correction to the individual log-likelihood ratios are applied in the test statistic to adjust for the number of parameters.

From the test results, we found the log-likelihood of the 17-SNP model was -2869 and the log-likelihood of the 9-logic tree interaction model was -2883 for 4684 observations. The value of the test statistic is 2227 and we can conclude that the interaction model is performing statistically significantly better than the 17-SNP main effects model (with p value = 0.00082).

To see if the logic regression model can differentiate between the cases and the controls graphically, we plotted the densities of the cross-validated log total odds ratios (OR) for the cases and controls in Figure 3.3. The total OR for the i^{th} person can be defined by, $OR_i = \prod_{j=1}^{9} OR_{ij}$, for the 9 logic trees. The cross-validated log total odds ratios had been obtained using a 10-fold cross-validation technique explained as follows.

1. We split the dataset into 10 roughly equal parts (using a stratified random sampling for cases and controls) and held out one part as the test set.

2. We fitted nine folds of data (training set) to the 9 tree-17 leaves logic regression model involving a search for the best model (based on minimum deviance of 30

logic regression model repeated with 30 different seeds) of the same size. Then we estimated the odds ratios corresponding to each tree.

3. We used the odds ratios obtained in Step 2 to calculate the total odds ratios for the test set data.

4. We repeated the process (Step 1-3) by treating each fold of data out of the 10 folds as a test set simultaneously. We obtained total odds ratios for the 10 sets of test data.

5. We plotted the density of the log total odds ratios for the test set data for cases and controls.

From the plot of the densities of log total OR for cases and controls we can observe that the logic regression model cannot differentiate between the cases from the controls very well. 28.7% of the cases have total log OR of more than 0 and 18.7% of the controls have total log OR of more than 0.

3.2 Two Stage Procedure to Check Interaction across Whole Genome

3.2.1 Analysis

The analysis discussed in this section was performed in a two stage procedure as described in Subsection 2.2.2: first stage involved in selecting a subset of SNPs and the

second stage involved in searching for SNP-interaction involving the set of SNPs using logic regression. The dataset we used to test this method cannot be shown and we have hidden all identifiable information such as the disease name. As this thesis is methodological in nature, we emphasize on developing methods and use the datasets to demonstrate their applications. For brevity, we call this dataset Dataset 1. The analysis was tested on the data of 706 cases and 514 controls and the genotyping of this dataset was carried out in Human610-Quad BeadChip (Illumina) platform.

First Stage

The first stage of the analysis involved filtering SNPs based on marginal association. The logic behind using marginal association as a filter for SNP- SNP interaction is that the SNP that interacts with other SNPs is likely to also display evidence of marginal association with the phenotype[26]. We performed logistic regression for each SNP of the 22 chromosomes of the Dataset 1 where the genotypes were coded as binary predictors. We compared each logistic regression model with a null model having no other predictors using a likelihood ratio test. The SNPs that had statistically significant effect based on marginal association (with p value less than 0.001) were selected to be the SNPs to enter the second stage for the test of interaction. In total, out of the 497242 SNPs, 843 statistically significant SNPs were selected based on marginal association to enter the next stage.

Second Stage

In the second stage, we performed logic regression analysis on the 843 SNPs together so that the SNPs from different chromosomes can interact with each other, as opposed to the gene-level analysis. We used a 10-fold cross validation technique to select the optimum number of leaves and trees for the logic regression. For the purpose of selecting the optimum number of trees and leaves, we explored a range of the number of trees from 1 to 9 and a range of the number of leaves from 1 to 25. The model for a 10-fold cross validation was run independently 30 times varying the seed of the random number generator at the beginning of the stochastic search of logic regression. The logic behind varying the seeds is to be able to search more broadly into the solution space and reduce the probability of converging to a local optimum. Since the search of the solution space is done stochastically by means of a Simulated Annealing algorithm, we fit the logic regression 30 times varying the initial random seed of the stochastic search. At the end of the 30 fitting processes, we keep the model with the lowest deviance among them.

3.2.2 Results

The resulting optimum combination of the numbers for the trees and leaves was 5 trees and 19 leaves, which were selected based on the minimum of the minimum prediction score produced from the 10-fold cross validation models generated across 30 different seeds. The 19 SNP shown in Table 3.4 are in the best logic regression model after analyzing all the marginally significant SNPs of Dataset 1. The table provides summary

of each SNP along with the percentage of those SNPs in cases and controls. The logic trees formed with these 19 SNPs are displayed in Table 3.5 along with the associated odds ratios. From this table 3.5, we can observe that the first logic tree is composed of five SNPs (rs2213953, rs17708991, rs9284844, rs12633500 and rs1999670) and the associated odds ratio of their interaction is 101.49. The second and fourth logic tree, which are composed of 2 SNPs each, have associated odds ratios of 0.26 and 0.24, respectively. The third logic tree composed of six SNPs with a complex form of interaction has associated odds ratio of 0.04. The fifth logic tree is composed of four SNPs with an associated odds ratio of 12.43.

We can observe from Table 3.4, that SNP rs1999670 (PPP1R14C gene) and SNP rs12633500 (LOC730109 gene) is highly prevalent in cases and SNP rs9284844 (SATB1 gene) is very rare in both cases and controls. We have found four SNPs/genes, namely, HLA-DRB1 (Chr 6p21.3), NOD2 (Chr 16q21), C13orf31 (Chr 13q14.11) and HLA-B (Chr 6p21.3) to be overlapping with the genes found in the other GWASs done for the disease related to Dataset 1. Therefore, 15 of the 19 SNPs that we found are potentially novel susceptibility genes for the phenotype of interest in Dataset 1 GWAS.

To check if the interaction model is performing better than a model that can be formed with the main effects of 19 SNPs, we used Clarke's non-nested model test[35,36] to compare the two models. Two models are non-nested if one model cannot be reduced to the other model by imposing a set of linear restrictions on the parameter vector or they are non-nested in terms of their functional forms. The interaction model and the 19 SNP main effects model will be non-nested as we cannot impose a set of linear

restriction on the parameter vector of one model to get another model. Usual tests to discriminate between models such as the likelihood ratio test cannot compare two models if they are not nested.

This test applies a modified paired sign test to the differences in the individual log-likelihoods from two nonnested models and determines whether or not the median log-likelihood ratio is statistically significantly different from zero. The test statistic is simply the number of positive differences in the individual log-likelihoods, which has a Binomial distribution with parameter n (number of observations) and $\theta = 0.5$. An average correction to the individual log-likelihood ratios are applied in the test statistic to adjust for the number of parameters.

From the test result, we found the log-likelihood of the 19-SNP model was -523 and the log-likelihood of the 5-logic tree interaction model was -526 for 1220 observations. The value of the test statistic is 303 and we can conclude that the interaction model is performing statistically significantly better than the 19-SNP main effects model (with p value = 2×10^{-16}).

To see if the logic regression model can differentiate between the cases and the controls graphically, we plotted the densities of the cross-validated log total odds ratios (OR) for the cases and controls in Figure 3.4. The total OR for the i^{th} person can be defined by, $OR_i = \prod_{j=1}^{5} OR_{ij}$, for the 5 logic trees. The cross-validated log total odds ratios had been obtained using a 10-fold cross-validation technique explained as follows.

1. We split the dataset into 10 roughly equal parts (using a stratified random sam-

pling for cases and controls) and held out one part as the test set.

2. We fitted nine folds of data (training set) to the 5 tree-19 leaves logic regression model involving a search for the best model (based on minimum deviance of 30 logic regression model repeated with 30 different seeds) of the same size. Then we estimated the odds ratios corresponding to each tree.

3. We used the odds ratios obtained in Step 2 to calculate the total odds ratios for the test set data.

4. We repeated the process (Step 1-3) by treating each fold of data out of the 10 folds as a test set simultaneously. We obtained total odds ratios for the 10 sets of test data.

5. We plotted the density of the log total odds ratios for the test set data for cases and controls.

From the plot of the densities of log total OR for cases and controls we can observe how much the logic regression model can differentiate the cases from the controls. 87.0% of the cases have total log OR of more than 0 and 52.1% of the controls have total log OR of more than 0.

3.3 Exploration of the SNP Pairs

3.3.1 A Proposed Analysis for SNP Pairs

We explored another approach to explore SNP-interaction in GWAS data which is based on SNP pairs. The basic idea here is to explore the SNP pairs such that we consider a pair of SNPs with genotype patterns that are only observed in cases with a sufficient frequency and no control has the specific genotype patterns. To test this idea we used the Dataset 1 explained in Subsection 3.2.1 with 706 cases and 514 controls. We only used the data of 843 SNPs that we filtered based on statistically significant marginal association as explained in the first stage of analysis in Subsection 3.2.1. In this data, we searched for the SNP pairs which contain a genotype where at least 13 cases have that genotype and none of the control does. Genotypes with at least 13 cases out of 706 cases versus 0 controls out of 514 controls would be fairly rare corresponding to P-value equals to 0.001 based on Fisher's exact test.

3.3.2 Results

The possible combination of SNP pairs using 843 SNPs yielded 354903 SNP pairs. We looked for the SNP pairs among 354903 pairs that have genotypes such that at least 13 cases have that genotype and none of the control does. We found 8348 such SNP pairs; 8268 SNP pairs having one such genotype pair where at least 13 cases have that genotype pair and none of the control does and 80 SNP pairs having two such genotypes pairs where at least 13 cases have those genotypes pairs and none of the control does.

In total we found 8428 genotype pairs to analyze further. Among the 8428 SNP pairs, we have analyzed the pattern among the cases' being positive to any of these pairs. To validate our findings, we have mapped the SNPs with another dataset of the same phenotype of interest, namely, Dataset 2, and analyzed only the SNP pairs that are evaluable both in Dataset 1 and Dataset 2.

From Tables 3.6 and 3.7 we can observe that chromosome 3 and 6 has the highest number of SNPs and SNP-pairs among the SNPs selected based on the criteria explained in Subsection 3.3.1. We analyzed the pattern of SNP pair positives among the cases (706) of Dataset 1. By SNP pair positives we mean cases' being positive for one or more SNP pairs. Table 3.8 shows the distribution of the cases with one or more SNP pairs being positive. From Table 3.8 we can see that at least three or more of these SNP pairs are present in 93.77% of cases in Dataset 1. Only 24 (3.40%) of the 706 cases do not have any of these SNP pairs. About 50% of the cases have at least 75 or more of these SNP pairs.

Mapping the SNPs of Dataset 1 with Dataset 2

To validate the SNP pairs obtained from Dataset 1, we tested this method on Dataset 2. Both the data used different platform for genotyping and thus we had to map the SNPs between both the data to see how many evaluable SNP pairs both data sets have in common. A total of 1848 evaluable SNP pairs from Dataset 2 have been found which is about 22% of the Dataset 1 SNP pairs. The equivalence of the genotypes was mapped between the Dataset 1 and Dataset 2.

The pattern of positives among the Dataset 2 cases and controls are explored and displayed in Table 3.9 and Table 3.10. Total number of Dataset 2 cases and controls are 285 and 124, respectively. We can see in Table 3.9 that none of these SNP-pairs are present in 105 (36.84%) cases out of the 285 cases. At least three or more of these SNP pairs are present in 53.33% of the cases in Dataset 2. We can see in Table 3.10 that none of these SNP-pairs are present in 51 (41.13%) controls out of the 124 controls. At least three or more of these SNP pairs are present in 51.61% of the controls.

In Dataset 2, the SNP pairs can be considered as "working well in Dataset 2" that are present in: 8 cases and 0 controls (6 pairs); 9 cases and 0 controls (12 pairs); 12 cases and 0 controls (4 pairs); or 16 cases and 2 controls (2 pairs) with Fisher's exact test p-value less than 0.05, as presented in Table 3.11.

Factor Analysis of the SNP Pairs Being Positive in Dataset 1 Cases

The goal of this analysis was to check if the findings from Dataset 1 can predict the SNP pairs that work well in Dataset 2. We performed factor analysis on the data of the SNP pairs being positive in cases obtained from Dataset 1 that were also evaluable in Dataset 2. The numbers of SNP pairs in Dataset 1 that are evaluable in Dataset 2 were 1848. There were SNPs in these 1848 SNP pairs that belonged to multiple genes (because they are between genes) and therefore there were perfectly correlated SNP pairs among the 1848 pairs. We removed those SNP pairs with perfect correlation which resulted in 636 SNP pairs to be analysed.

We performed principal component factor analysis with varimax rotation to extract

factors that may indicate specific clustering of SNP pairs in the cases of the Dataset 1. Thirteen (13) factors have been extracted with the reasoning that beyond factor 13, none of the factors explain individually more than 1% of the variation. In the 13 factors, the items with factor loadings greater than 0.4 are kept. Table 3.12 shows the 13 factors extracted with eigen values, proportion of explained variation and the number of items for each factor. Among the 13 factors, we checked how many of them contain SNP pairs that worked in Dataset 2 and we found that only Factor 6 and 7 contain 5 SNP pairs that work well in Dataset 2 as well. The sixth column in this table shows the number of SNP pairs that work well in Dataset 2 for each factor.

Identifying Good Factors from Bad Factors Based on Dataset 1

To distinguish good factors from the bad factors based on the Dataset 1 and without the knowledge from the Dataset 2, we used the following methods. By good factor we mean the factors that are composed of SNP pairs that would work in Dataset 2 as well as Dataset 1 and by bad factor we mean the factors that are composed of SNP pairs that would not work in Dataset 2.

(A) Simple Counting Method:

The idea here is to see if the good factors tend to be more positive for cases which are positive for many SNP pairs. That is, cases that are positive for many SNP pairs would be "definitely cases". Factors that are positive for "definite cases" may be the good factors. We performed the following steps and the results are

presented in Table 3.13.

(a) We counted the number of SNP pairs being positive for each case of Dataset 1. We called this number as total positive count (TPC) of the case.

(b) For a given factor, we looked at the loadings of 636 SNP pairs and selected the top 20% pairs to be the highly loaded pairs for each factor. For example, Factor 1 had 136 pairs and we selected top 20% pairs of 136, i.e., the top 28 pairs.

(c) For a given factor, the sum of the TPCs of cases who are positive to at least one of the highly loaded pairs of the factor is calculated. We called this the STPC (sum of total positive count) of the factor.

(d) For a given factor, STPC is divided by the number of cases who are positive to at least one of the highly loaded pairs of the factor. We called this the MTPC (mean of total positive count) of the factor.

From the results of Table 3.13 we can see that the factors 1, 2, 4, 5, 7 and 10 have higher STPC and MTPC. However, we cannot clearly distinguish between good factors and bad factors. We have seen from Table 3.12 that Factors 6 and 7 are working well in Dataset 2, but this technique could not clearly differentiate them from the other factors.

(B) Double Factor Analysis:

We performed a double factor analysis to distinguish between good factors and

bad factors. The idea here is that the factor analysis done that obtained 13 factors from the pattern of cases' being positive for the SNP pairs in Dataset 1 would look at variations of "significant SNP pairs" in cases. That is, these SNP pairs are the the variables that are supposed to be signals that separate the cases from controls. However, many of these "signals" are not validated by Dataset 2 and may not be real signals: i.e., they are just the natural variation among the people (cases). We proposed performing step (a)-(c), as described below, where we check the correlation among the factor scores of the 13 factors, with the factor scores of the cases only, obtained from a second factor analysis of the "SNPs" of Dataset 1, by mixing the data of both the cases and controls. By performing step (a)-(c), we would get the patterns of natural variation among all people. If a factor from step (a) factor analysis correlates with a factor from step (c), then that is likely to be a natural variation and it may indicate that the SNP pairs of the 13 factors are not providing us with any special knowledge of clustering among the SNPs that separates cases from the controls. A factor from step (a) that has less correlation with factors from step (c) may be a good factor that is able to discriminate between cases and controls. Similar logic can be established for step (a) and (d) where we check the correlation among the 13 factor scores obtained from step (a) and the factor scores obtained from the factor analysis performed on the"SNPs" of Dataset 1, for cases only.

(a) Factor scores of 13 factors are derived for the Dataset 1's cases' being positive

for SNP pairs data (Table 3.12).

(b) From the 843 SNPs that we selected based on statistically significant marginal association explained in Subsection 3.2.1, we removed the perfectly correlated SNPs and obtained 516 SNPs for further analysis.

(c) A second factor analysis (Principal component factor analysis with varimax rotation) was run mixing cases and controls of the 516 SNPs of the Dataset 1 and the factor scores are derived. In total 5 factors were extracted in this analysis. Correlations of each factor score in (a) and each factor score in (c) among cases are calculated and presented in Table 3.14.

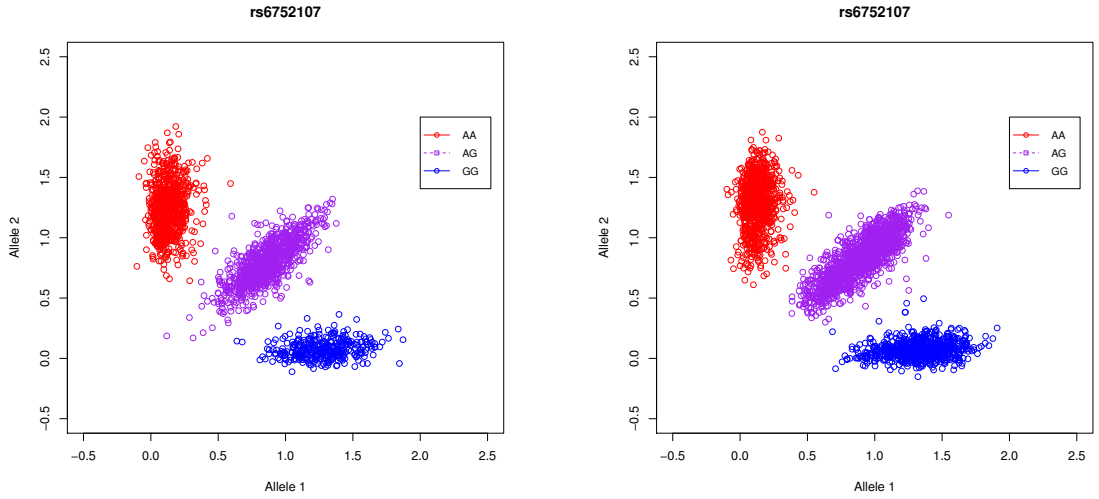
(d) Factor analysis (Principal component factor analysis with varimax rotation) is repeated with only the cases of the 516 SNPs of Dataset 1 and the factor scores are derived. In total, 4 factors were extracted in this analysis. Correlations of each factor score in (a) and each factor score in (d) among cases are calculated and reported in Table 3.15.

From the correlation matrices in Table 3.14 and 3.15, we can observe that all the factors in step (a) has very small correlation with the factors from step (c) and (d). That is, according to the logic applied for this method, all the factors can be considered as good factors. Therefore, this method could not differentiate Factor 6 and 7 from the rest of the factors, either.

Analysis of the Shared Pathways

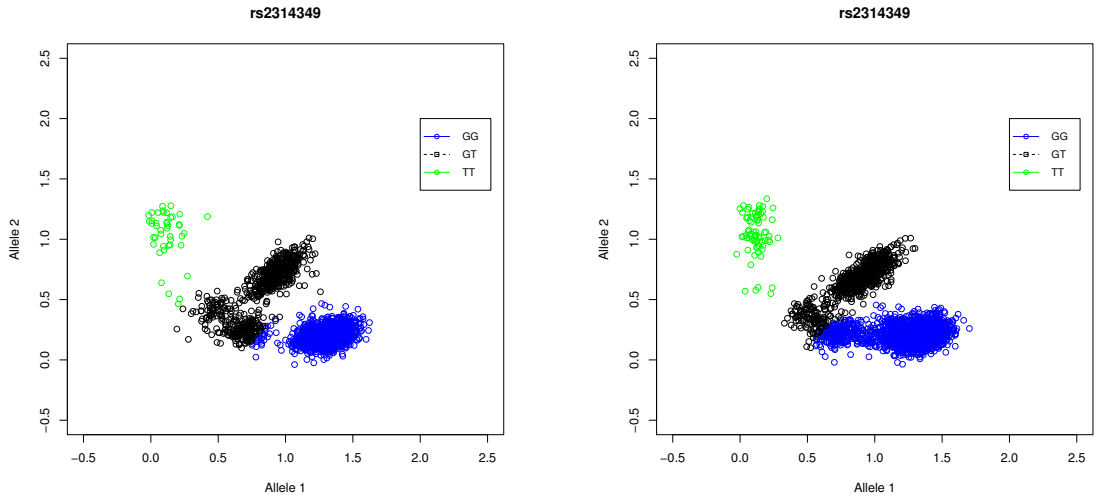
We have calculated the number of shared pathways between each SNP pair for the 636 SNP pairs using the methods developed in the work of Conrado Franco-Villalobos *et al.* entitled "Logic Regression Visualization Analysis of GWAS Provides New Insights into Shared Genetic Susceptibilities Among a Group of Diseases". We analysed the 636 SNP pairs to see the distribution of the shared pathways among them. From Table 3.16, we can observe that there are 6 SNP pairs among the 636 pairs that work well in Dataset 2 as well. We explored these 6 pairs in Dataset 2 and the number of being positive for those SNP pairs among Dataset 2 cases and controls are presented in Table 3.17. We also checked the number of Dataset 2 cases and controls being positive to any of the 6 pairs and the results are reported in Table 3.18 and 3.19. From these tables, we can see that 34 Dataset 2 cases have at least one of these SNP pairs and only two Dataset 2 controls have at least one of these SNP pairs.

We mapped these 6 pairs with the factors obtained from the factor analysis shown in Table 3.12 and found that 4 of these pairs belong to Factor 6 and 1 of these factors belong to Factor 7. The factor loadings corresponding to these pairs are displayed in Table 3.20 and only the factor loadings greater than 0.4 are kept. Table 3.21 provides the SNP, gene and chromosome information of the 6 pairs obtained from the shared pathway analysis. To our knowledge, none of these genes or SNPs had been previously identified to have any association with the phenotype of interest for Dataset 1 and Dataset 2.



(a) Cases (b) Controls

Figure 3.1: Intensity plot for SNP-rs6752107 (high quality marker) in WTCCC CD GWAS data.



(a) Cases (b) Controls

Figure 3.2: Intensity plot for SNP-rs2314349 (low quality marker) in WTCCC CD GWAS data.

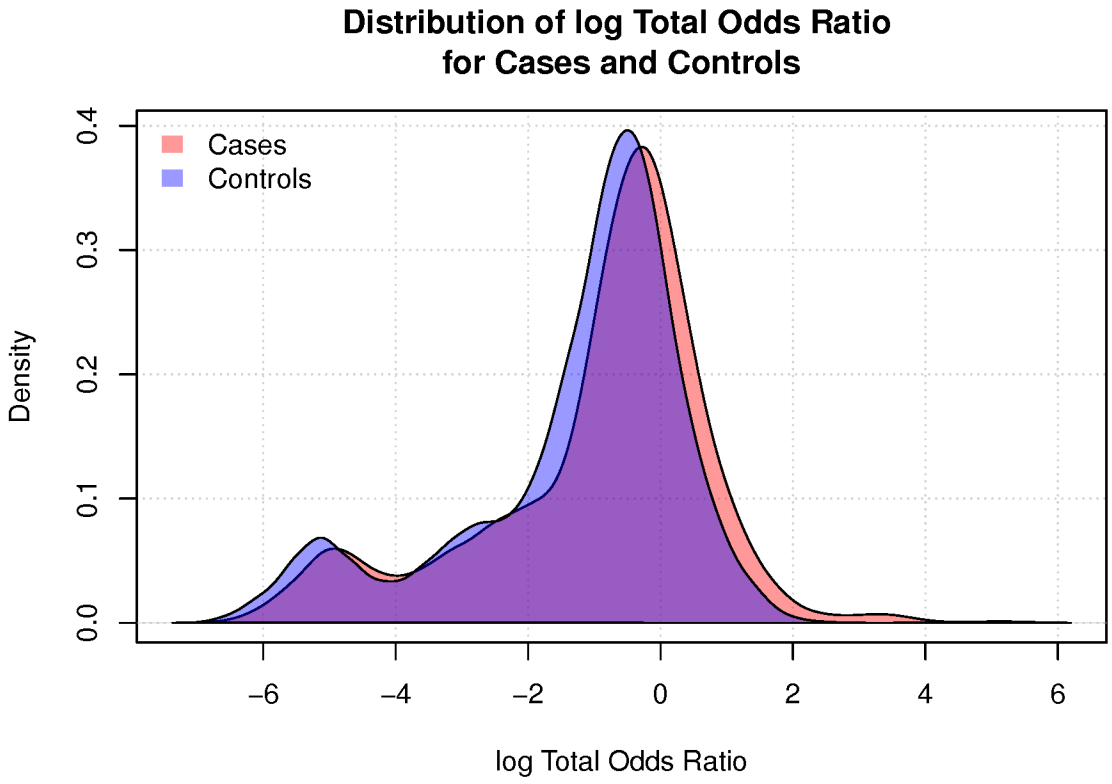


Figure 3.3: Densities of the cross-validated log Total Odds Ratio for the cases and controls discriminated by Logic Regression model for Crohn's Disease GWAS .

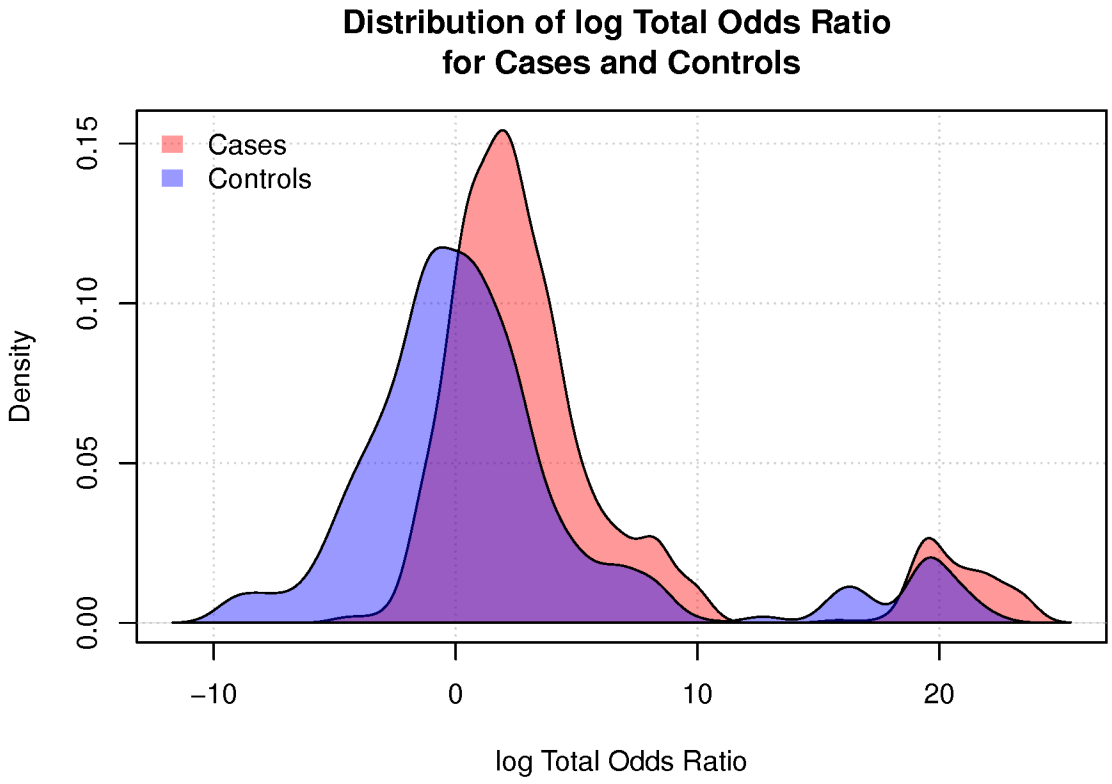


Figure 3.4: Densities of the cross-validated log Total Odds Ratio for the cases and controls discriminated by Logic Regression model for Dataset 1 GWAS .

Table 3.1: **Genetic information of the 17 SNPs obtained from the Logic Regression for CD GWAS data**

SNP	Gene	Chromosome	Case (%)	Control (%)
rs10427270	CERKL	2q31.3	183 (10.47%)	271 (9.23%)
rs723713	NRXN1	2p16.3	1053 (60.24%)	1589 (54.12%)
rs10761659	ZNF365	10q21.2	293 (16.76%)	631 (21.49%)
rs9858542	BSN	3p21.31	217 (12.41%)	219 (7.46%)
rs6752107	ATG16L1	2q37.1	633 (36.21%)	760 (25.89%)
rs13361189	MST150	5q33.1	1410 (80.66%)	2537 (86.41%)
rs7234029	PTPN2	18p11.3-p11.2	1111 (63.56%)	2083 (70.95%)
rs6500315	NKD1	16q12.1	53 (3.03%)	161 (5.48%)
rs11805303	IL23R	1p31.3	1093 (62.53%)	1552 (52.86%)
rs7515029	C1orf141	1p31.3	1674 (95.77%)	2680 (91.28%)
rs7807268	C7orf33	7q36.1	397 (22.71%)	858 (29.22%)
rs888775	WWC1	5q34	1725 (98.68%)	2936 (100%)
rs6674713	PTCHD2	1p36.22	1736 (99.31%)	2934 (99.93%)
rs17221417	NOD2	16q21	1502 (85.93%)	2680 (91.28%)
rs11957215	PTGER4	5p13.1	972 (55.61%)	1340 (45.64%)
rs10883371	NKX2-3	10q24.2	1239 (70.88%)	2285 (77.83%)
rs17234657	PTGER4	5p13.1	1179 (67.45%)	2256 (76.84%)

Table 3.2: **Logic trees and the associated odds ratios of the 9 tree Logic Regression model for CD**

Logic Trees	Odds Ratios	Case (%)	Control (%)
((rs723713 and (not rs10761659)) or rs10427270)	1.53	993 (56.81%)	1377 (46.9%)
(rs9858542 or rs6752107)	1.75	768 (43.94%)	918 (31.27%)
((not rs13361189) or (not rs7234029))	1.52	851 (48.68%)	1121 (38.18%)
(rs6500315 or (not rs11805303))	0.66	686 (39.24%)	1479 (50.37%)
(rs7515029 and (not rs7807268))	1.51	1295 (74.08%)	1894 (64.51%)
((not rs888775) or (not rs6674713))	27.24	34 (1.95%)	2 (0.07%)
rs17221417	0.59	1502 (85.93%)	2680 (91.28%)
(rs11957215 or (not rs10883371))	1.49	1202 (68.76%)	1688 (57.49%)
(not rs17234657)	1.49	569 (32.55%)	680 (23.16%)

Table 3.3: **Comparison of our results with single-SNP WTCCC and meta-analysis results based on overlapping SNP, gene or chromosomal location for CD**

SNP	Gene	Chromosomal location	Single-SNP WTCCC[11] (strength of association)	Meta Analysis[12]
rs10427270	CERKL	2q31.3	-	-
rs723713	NRXN1	2p16.3	-	Chromosomal location
rs10761659	ZNF365	10q21.2	Gene (strong)	SNP
rs9858542	BSN	3p21.31	Gene (strong)	Gene
rs6752107	ATG16L1	2q37.1	Gene (strong)	Gene
rs13361189	MST150	5q33.1	Gene (strong)	Chromosomal location
rs7234029	PTPN2	18p11.3 -p11.2	Gene (strong)	Gene
rs6500315	NKD1	16q12.1	Gene (strong)	Chromosomal location
rs11805303	IL23R	1p31.3	SNP/Gene (strong)	Gene
rs7515029	C1orf141	1p31.3	Chromosomal location (strong)	Chromosomal location
rs7807268	C7orf33	7q36.1	SNP (moderate)	-
rs888775	WWC1	5q34	-	-
rs6674713	PTCHD2	1p36.22	-	Chromosomal location
rs17221417	NOD2	16q21	SNP/Gene (strong)	Gene
rs11957215	PTGER4	5p13.1	Chromosomal location (strong)	Gene
rs10883371	NKX2-3	10q24.2	Gene (strong)	Gene
rs17234657	PTGER4	5p13.1	SNP/Gene (strong)	Gene

Table 3.4: **Genetic information of the 19 SNPs obtained from the Logic Regression of Dataset 1**

SNP	Gene	Chromosome	Case (%)	Control (%)
rs1999670	PPP1R14C	6q24.3-q25.3	706 (100%)	464 (90.27%)
rs2213953	LOC100287927	22q11.22	704 (99.72%)	477 (92.8%)
rs17708991	LOC100129711	6q15	1 (0.14%)	12 (2.33%)
rs9284844	SATB1	3p23	0 (0%)	8 (1.56%)
rs12633500	LOC730109	3q25.33	706 (100%)	506 (98.44%)
rs9271348	HLA-DRB1	6p21.3	218 (30.88%)	269 (52.33%)
rs9302752	NOD2	16q21	485 (68.7%)	253 (49.22%)
rs13381553	LOC441806	18p11.32	698 (98.87%)	478 (93%)
rs7945327	TRPM5	11p15.5	705 (99.86%)	502 (97.67%)
rs2426714	RBM38	20q13.31	698 (98.87%)	470 (91.44%)
rs9551445	LOC100287114	13q12.11	705 (99.86%)	490 (95.33%)
rs6846231	LOC100288073	4q32.3	705 (99.86%)	488 (94.94%)
rs17132673	DTWD2	5q23.1	701 (99.29%)	482 (93.77%)
rs3764147	C13orf31	13q14.11	218 (30.88%)	253 (49.22%)
rs9264904	HLA-B	6p21.3	297 (42.07%)	154 (29.96%)
rs11752822	LOC100289273	6q27	650 (92.07%)	509 (99.03%)
rs9658807	BATF3	1q32.3	642 (90.93%)	507 (98.64%)
rs6556066	NKX2-5	5q34	663 (93.91%)	509 (99.03%)
rs2676870	C3orf21	3q29	628 (88.95%)	500 (97.28%)

Table 3.5: **Logic trees and the associated odds ratios of the 5-tree Logic Regression model for Dataset 1**

Logic Trees	Odds Ratios	Case (%)	Control (%)
(((rs2213953 and (not rs17708991)) and ((not rs9284844) and rs12633500)) and rs1999670)	101.49	703 (99.58%)	407 (79.18%)
(rs9271348 or (not rs9302752))	0.26	367 (51.98%)	401 (78.02%)
((((not rs2426714) or (not rs9551445)) or ((not rs6846231) or (not rs17132673))) or ((not rs13381553) or (not rs7945327)))	0.04	24 (3.4%)	163 (31.71%)
(rs3764147 and (not rs9264904))	0.24	110 (15.58%)	176 (34.24%)
(((not rs11752822) or (not rs9658807)) or ((not rs6556066) or (not rs2676870)))	12.43	203 (28.75%)	28 (5.45%)

Table 3.6: **Frequencies of SNPs in the 8348 high risk SNP pairs from each chromosome**

Chromosome	Number of SNPs within this chromosome (%)	Chromosome	Number of SNPs within this chromosome (%)
1	1353 (8.1%)	12	685 (4.1%)
2	561 (3.4%)	13	675 (4%)
3	3100 (18.6%)	14	863 (5.2%)
4	472 (2.8%)	15	304 (1.8%)
5	968 (5.8%)	16	406 (2.4%)
6	2746 (16.4%)	17	257 (1.5%)
7	444 (2.7%)	18	451 (2.7%)
8	523 (3.1%)	19	62 (0.4%)
9	905 (5.4%)	20	330 (2%)
10	588 (3.5%)	21	146 (0.9%)
11	383 (2.3%)	22	474 (2.8%)

Table 3.7: **Top 20 most frequent chromosome pairs in the 8348 high risk SNP pairs**

Chromosome pair	Frequency of appearing among 8348 SNP pairs (%)	Chromosome pair	Frequency of appearing among 8348 SNP pairs (%)
(3, 6)	590 (7.1%)	(3, 12)	145 (1.7%)
(1, 6)	254 (3%)	(2, 3)	141 (1.7%)
(1, 3)	226 (2.7%)	(3, 8)	134 (1.6%)
(6, 6)	217 (2.6%)	(3, 13)	123 (1.5%)
(6, 9)	189 (2.3%)	(3, 14)	114 (1.4%)
(3, 5)	183 (2.2%)	(3, 10)	108 (1.3%)
(3, 9)	180 (2.2%)	(3, 7)	108 (1.3%)
(6, 14)	177 (2.1%)	(3, 4)	104 (1.2%)
(5, 6)	168 (2%)	(3, 18)	98 (1.2%)
(3, 3)	149 (1.8%)	(6, 12)	98 (1.2%)

Table 3.8: **Frequency (%) of cases with one or more SNP pairs in Dataset 1**

Number of SNP pairs	Frequency (%)
0	24 (3.40%)
3 or more	662 (93.77%)
5 or more	644 (91.22%)
10 or more	606 (84.84%)
20 or more	533 (75.50%)
30 or more	480 (67.99%)
40 or more	435 (61.61%)
50 or more	408 (57.79%)
75 or more	351 (49.72%)
100 or more	301 (42.63%)
300 or more	135 (19.12%)
500 or more	73 (10.34%)
1000 or more	25 (3.54%)
1500 or more	6 (0.85%)

Table 3.9: **Frequency (%) of subjects with one or more SNP pairs in Dataset 2 cases**

Number of SNP Pairs	Frequency (%)
0	105 (36.84%)
3 or more	152 (53.33%)
5 or more	131 (45.96%)
10 or more	111 (38.95%)
20 or more	84 (29.47%)
30 or more	64 (22.46%)
40 or more	42 (14.74%)
50 or more	30 (10.53%)
75 or more	11 (3.86%)
100 or more	7 (2.46%)
300 or more	1 (0.35%)

Table 3.10: **Frequency (%) of subjects with one or more SNP pairs in Dataset 2 controls**

Number of SNP Pairs	Frequency (%)
0	51 (41.13%)
3 or more	64 (51.61%)
5 or more	51 (41.13%)
10 or more	40 (32.26%)
20 or more	22 (17.74%)
30 or more	15 (12.10%)
40 or more	11 (8.87%)
50 or more	8 (6.45%)
75 or more	1 (0.81%)
100 or more	1 (0.81%)

Table 3.11: **Distribution of the SNP pairs in Dataset 2 among cases and controls**

SNP pair combination	Frequency (%)	SNP pair combination	Frequency (%)
0 case, 0 control	165 (8.93%)	6 case, 2 control	12 (0.65%)
0 case, 1 control	57 (3.08%)	6 case, 3 control	2 (0.11%)
0 case, 2 control	40 (2.16%)	6 case, 4 control	8 (0.43%)
0 case, 3 control	7 (0.38%)	6 case, 5 control	2 (0.11%)
1 case, 0 control	220 (11.9%)	7 case, 0 control	6 (0.32%)
1 case, 1 control	101 (5.47%)	7 case, 1 control	5 (0.27%)
1 case, 2 control	56 (3.03%)	7 case, 2 control	14 (0.76%)
1 case, 3 control	14 (0.76%)	7 case, 3 control	4 (0.22%)
1 case, 4 control	4 (0.22%)	7 case, 4 control	10 (0.54%)
1 case, 5 control	4 (0.22%)	7 case, 5 control	8 (0.43%)
2 case, 0 control	218 (11.8%)	8 case, 0 control	6 (0.32%)
2 case, 1 control	102 (5.52%)	8 case, 1 control	8 (0.43%)
2 case, 2 control	90 (4.87%)	8 case, 2 control	4 (0.22%)
2 case, 3 control	14 (0.76%)	8 case, 3 control	5 (0.27%)
2 case, 4 control	10 (0.54%)	8 case, 4 control	9 (0.49%)
3 case, 0 control	64 (3.46%)	8 case, 5 control	3 (0.16%)
3 case, 1 control	73 (3.95%)	9 case, 0 control	12 (0.65%)
3 case, 2 control	80 (4.33%)	9 case, 1 control	4 (0.22%)
3 case, 3 control	10 (0.54%)	9 case, 2 control	6 (0.32%)
3 case, 4 control	4 (0.22%)	9 case, 3 control	8 (0.43%)
4 case, 0 control	68 (3.68%)	9 case, 4 control	7 (0.38%)
4 case, 1 control	69 (3.73%)	9 case, 7 control	2 (0.11%)
4 case, 2 control	54 (2.92%)	10 case, 2 control	11 (0.6%)
4 case, 3 control	7 (0.38%)	10 case, 3 control	14 (0.76%)
4 case, 4 control	2 (0.11%)	10 case, 5 control	7 (0.38%)
5 case, 0 control	9 (0.49%)	10 case, 7 control	2 (0.11%)
5 case, 1 control	33 (1.79%)	11 case, 1 control	4 (0.22%)
5 case, 2 control	17 (0.92%)	11 case, 2 control	2 (0.11%)
5 case, 3 control	8 (0.43%)	12 case, 0 control	4 (0.22%)
5 case, 4 control	16 (0.87%)	12 case, 2 control	5 (0.27%)
5 case, 5 control	4 (0.22%)	12 case, 3 control	5 (0.27%)
5 case, 7 control	1 (0.05%)	13 case, 2 control	4 (0.22%)
6 case, 0 control	6 (0.32%)	14 case, 3 control	2 (0.11%)
6 case, 1 control	14 (0.76%)	16 case, 2 control	2 (0.11%)

Table 3.12: **Eigen values, proportion of explained variation and the number of items in each factor with items that work in Dataset 2**

Factors	Eigen Value	Proportion Explained	Cumulative Proportion Explained	Number of Items	Number of Pairs Working in Dataset 2
1	102.80	0.16	0.16	136	0
2	84.67	0.13	0.29	156	0
3	20.82	0.03	0.33	25	0
4	19.17	0.03	0.36	54	0
5	18.19	0.03	0.39	22	0
6	15.92	0.03	0.41	27	4
7	15.25	0.02	0.44	33	1
8	13.56	0.02	0.46	23	0
9	9.83	0.02	0.47	10	0
10	8.69	0.01	0.49	14	0
11	7.90	0.01	0.50	8	0
12	7.69	0.01	0.51	12	0
13	6.61	0.01	0.52	9	0

Table 3.13: **Sum and mean total positive count for 13 factors**

Factor	Number of Items	Number of highly loaded pairs	STPC (sum of total positive counts)	Number of cases being positive for at least one of the highly loaded pairs	MTPC (mean of total positive counts)
1	136	28	2300	21	109.52
2	156	32	3289	33	99.67
3	25	5	565	19	29.74
4	54	11	2854	53	53.85
5	22	5	671	18	37.28
6	27	6	775	26	29.81
7	33	7	1277	33	38.70
8	23	5	712	24	29.67
9	10	2	365	14	26.07
10	14	3	881	22	40.05
11	8	2	516	15	34.40
12	12	3	681	31	21.97
13	9	2	444	30	14.80

Table 3.14: **Correlation between factor scores of the SNP pairs of Dataset 1 cases and original SNP data of Dataset 1 for cases (using case-control mix data)**

	Factor 1	Factor 2	Factor 3	Factor 4	Factor 5
Factor 1	-0.04	-0.05	0.06	-0.04	-0.01
Factor 2	-0.06	-0.00	0.11	-0.10	0.08
Factor 3	0.04	0.07	-0.04	0.04	0.01
Factor 4	0.04	0.04	0.06	-0.18	0.06
Factor 5	0.01	-0.00	-0.01	-0.02	0.00
Factor 6	-0.01	-0.02	0.03	0.01	0.10
Factor 7	-0.03	-0.02	0.02	-0.01	0.01
Factor 8	-0.02	0.10	0.04	0.00	0.00
Factor 9	-0.07	0.02	0.08	-0.01	-0.01
Factor 10	-0.06	0.01	0.07	-0.10	0.02
Factor 11	-0.00	0.06	0.04	-0.07	0.00
Factor 12	0.20	0.10	-0.03	-0.06	0.07
Factor 13	0.19	-0.03	-0.11	-0.06	0.04

Table 3.15: **Correlation between factor scores of the SNP pairs of Dataset 1 cases and original SNP data of Dataset 1 for cases (using only cases' data)**

	Factor 1	Factor 2	Factor 3	Factor 4
Factor 1	-0.03	0.02	-0.08	-0.02
Factor 2	-0.06	-0.04	-0.16	-0.05
Factor 3	0.04	-0.05	0.08	0.02
Factor 4	0.03	-0.07	-0.15	-0.14
Factor 5	0.01	-0.00	-0.01	-0.01
Factor 6	-0.01	0.02	-0.01	0.04
Factor 7	-0.03	0.01	-0.02	0.00
Factor 8	-0.02	-0.11	-0.00	0.03
Factor 9	-0.07	-0.04	-0.07	0.03
Factor 10	-0.06	-0.04	-0.12	-0.09
Factor 11	-0.00	-0.08	-0.06	-0.03
Factor 12	0.20	-0.05	0.00	0.02
Factor 13	0.19	0.08	0.04	-0.04

Table 3.16: **Distribution of the number of shared pathways by SNP pairs that worked in Dataset 2**

Number of shared pathways	SNP pairs that do not work well in Dataset 2 (%)	SNP pairs that work well in Dataset 2 (%)
0	408 (98.79%)	5 (1.21%)
1	86 (98.85%)	1 (1.15%)
2 and more	136 (100%)	0 (0%)

Table 3.17: **Number of being positive in Dataset 2 cases and controls for the 6 pairs that are working well**

Pairs Working Well in Dataset 2	Number of Positives in Dataset 2 Cases	Number of Positives in Dataset 2 Controls
1	8	0
2	9	0
3	9	0
4	8	0
5	16	2
6	9	0

Table 3.18: **Number of being positive to any of the 6 pairs among Dataset 2 cases**

Number of being positive to any of the 6 pairs	Frequency
0	251
1	23
2	1
3	6
4	4
Total Dataset 2 case	285

Table 3.19: **Number of being positive to any of the 6 pairs among Dataset 2 controls**

Number of being positive to any of the 6 pairs	Frequency
0	122
1	2
Total Dataset 2 control	124

Table 3.20: **Factor loadings of the 6 pairs (loadings greater than 0.4 are kept)**

Pairs	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13
1						0.66							
2						0.83							
3						0.75							
4						0.74							
5													
6							0.63						

Table 3.21: **Details of the SNP, gene and chromosome information of the 6 pairs)**

Pairs	SNP pair	Chromosome pair	Gene pair	Number of shared pathways
1	(rs7595482, rs1903524)	(2, 4)	(FAM82A1, LOC100288304)	0
2	(rs1903524, rs2200370)	(4, 15)	(LOC100288304, ATP10A)	1
3	(rs1903524, rs1109400)	(4, 20)	(LOC100288304, C20orf151)	0
4	(rs1903524, rs16994111)	(4, 20)	(LOC100288304, ESF1)	0
5	(rs6877393, rs6953592)	(5, 7)	(MCTP1, DKFZp564N2472)	0
6	(rs9347874, rs12607401)	(6, 18)	(LOC100289273, FLJ44881)	0

Chapter 4

Discussion and Conclusion

4.1 Review of the Objective and Methods

Analyses of GWAS in majority of the studies have tended to focus on single SNP analysis without the consideration of interaction among the SNPs that has more biological plausibility. The major objective of this thesis was to explore interaction among many genetic variants that may be associated with the disease phenotype. We have used different approaches to analyse GWAS data that incorporates interaction with the intention to explain missing heritability.

4.1.1 Methods Based on Logic Regression

In our first method where we performed logic regression across the whole genome, as explained in Section 3.1, we used a selected set of the top genes[27] to reduce the dimension of the space to search for SNP interaction. False positive results are a

common concern in GWAS due to the large number of tests performed. Our interaction analysis was performed on the SNPs after removing the SNPs that were found likely to be the results of genotyping error. One limitation of our approach was that we performed the check for genotyping error after we fixed our final model size using 10-fold cross validation to reduce the computational cost. We recognize that if we had checked and removed all the SNPs with potential genotyping errors first, the model size may have been changed.

Despite the limitation of our approach, we explored the possible interactions across the whole genome and our results illustrate the power of the logic-regression-based GWAS analysis to uncover new genetic susceptibilities and explain to a greater extent the CD genetics. In view of the standard approach of assessing the marginal effects of single SNPs one at a time, the method of logic regression proposed here provides a clear advance over single-SNP analysis. It can search for more biologically-plausible forms of SNP effects (incorporating interaction of redundancy and concurrence) with greater degrees of association indicated by appreciably larger values of odds ratios. In our analysis of CD GWAS data, we discovered 17 SNPs using this method and 15 of those genes were previously identified in different studies[11,12].

In the second method that we proposed, as explained in Section 3.2, we adopted the filtering technique[26] to select variants based on statistically significant marginal association to reduce the dimension of the search space for the interaction phase. One limitation of using SNPs filtered based on marginal association to test for interaction is that it would yield limited power if the form of the interaction is qualitative, i.e.,

the subgroup effects may cancel out when averaged[26]. We performed logic regression analysis across the whole genome of Dataset 1 using the filtered SNPs only and obtained a 19-SNP interaction model. We found four genes, namely HLA-DRB1, NOD2, C13orf31 and HLA-B, that have been already found in other GWASs for the phenotype of interest in Dataset 1. To our knowledge, the other 15 SNPs/genes of the 19 SNPs that we found are potentially novel susceptibility genes for the GWAS of the phenotype of interest for Dataset 1 (PPP1R14C, LOC100287927, LOC100129711, SATB1, LOC730109, LOC441806, TRPM5, RBM38, LOC100287114, LOC100288073, DTWD2, LOC100289273, BATF3, NKX2-5, C3orf21). However, these findings should be validated in order to rule-out the possibility of spurious associations due to population stratification or genotyping errors.

4.1.2 Method Based on SNP-Pair Analysis

The last method that we proposed is based on SNP pair analysis, as explained in Section 3.3. We used the filtered SNPs explained in previous method for Dataset 1 and explored the SNP pairs with genotypes that are only observed in cases with a sufficient frequency and no control has the specific genotypes. We have analyzed the pattern of being positive for any of those SNP pairs among the cases of Dataset 1 using factor analysis. A validation analysis has been performed on the SNP-pairs of Dataset 2 that were mapped with Dataset 1. Both the data have been generated through different platforms and we only had to use the data that were matched between the two data sets, which substantially reduced the number of available SNPs in the analysis. We

found 2 factors containing 5 SNP pairs that are present in the cases only and also work on Dataset 2.

We discussed two methods to distinguish the factors that would work in Dataset 2 and the ones that would not. They were unable to distinguish between factors that would work and would not. We have found 6 SNP pairs that work both in Dataset 1 and Dataset 2 based on the information of the number of shared pathways each SNP-pair has. Five of these SNP pairs are contained in the 2 factors that work in both Dataset 1 and Dataset 2. To our knowledge, the genes obtained from this SNP pair analysis (FAM82A1, LOC100288304, ATP10A, C20orf151, ESF1, MCTP1, DKFZp564N2472, LOC100289273, FLJ44881) have not been previously found to be associated with the disease of interest for these datasets.

4.2 Future Work

Further research goals can be established based on the work presented in this thesis for more complex diseases. Additionally, the findings of this study should be validated with other studies to eliminate the possibilities of spurious associations. Specifically, some of the scopes of further research would be:

- To conduct validation studies on the methodologies presented in this paper.
- To develop methods of classification to distinguish factors that can clearly separate the cases from the controls in GWAS.

- To examine each allele (or haplotype) for exploring the same interaction idea by logic regression.
- To develop other methods of SNP set selection other than filtering based on marginal association or candidate gene approach.
- To develop techniques for mapping GWASs of different diseases with different genotyping platforms.
- To develop methodologies that would help to find more specific solution to the "missing heritability" problem.

4.3 Conclusion

Increasing attention has been paid recently to interaction based analysis of GWAS. Such SNP-SNP interaction analysis could provide further valuable insights to understand the mechanism of the genetic variation which might modify the disease risks. The findings of this thesis work might help uncover the mechanism of many diseases that have not yet been found based on the patterns of the SNP interactions. We did not find a satisfactory solution to the missing heritability problem through this thesis work; however, in this work we showed the importance of considering SNP interactions and their exploration in considering genetic contributions of disease etiology, prevention and treatment.

Appendix A

R Codes

A.1 Logic Regression across Whole Genome

```
#####################################
#######CHECK FOR GENOTYPING ERROR########
#####################################

#FOR CASE

for (i in 3:3){
        #Load list of SNPs to plot, first column should be
           SNP, second column chromosome location
        listToLook=read.table("ListToLook.txt",sep="\t",
           header=FALSE)
        #Get the SNPs corresponding to the ith Chromosome
        ListToLook=listToLook[listToLook[,2]==i,1]

        #Read the genotype file
        infile=paste("Affx_20070205fs1_gt_IBD_0",i,".txt",
           sep="")
        gene=read.table(infile,header=FALSE)

        #Get the genotype for each SNP
        keep=which(as.character(gene[,1]) %in% as.character
           (ListToLook))
        genoCases=gene[keep,c(1,3)]
```

```
#Remove genotype file
rm(gene)

#Read the signal file
infile=paste("Affx_20070205f_signal_CD_0",i,".txt",
   sep="")
gene=read.table(infile,header=TRUE)

#Get the signal for each SNP
keep=which(as.character(gene[,2]) %in% as.character
   (ListToLook))
allelesCases=gene[keep,]

#Remove genotype file
rm(gene)
for (j in 1:length(ListToLook)){
        #Get the Allele #1 and Allele #2 for each
           SNP
        allele1=allelesCases[allelesCases[,2]==as.
           character(ListToLook[j]),seq(6,dim(
           allelesCases)[2],by=2)]
        allele2=allelesCases[allelesCases[,2]==as.
           character(ListToLook[j]),seq(7,dim(
           allelesCases)[2],by=2)]

        #Get the genotype for each SNP
        genotype=genoCases[genoCases[,1]==as.
           character(ListToLook[j]),2]
        genotypesort=sort(unique(genotype))
        color<-c('color1','color2','color3')

        if(length(unique(genotype))==1){

        for(k in 1:1){
               if(genotypesort[k]=='AA'){color[k]<-
                  "red"}
               else if(genotypesort[k]=='TT'){color
                  [k]<-"green"}
               else if(genotypesort[k]=='CC'){color
                  [k]<-"yellow"}
               else if(genotypesort[k]=='GG'){color
                  [k]<-"blue"}
               else if(genotypesort[k]=='AT'){color
                  [k]<-"orange"}
               else if(genotypesort[k]=='AC'){color
                  [k]<-"grey"}
```

```
        else if(genotypesort[k]=='AG'){color
           [k]<-"purple"}
        else if(genotypesort[k]=='CT'){color
           [k]<-"pink"}
        else if(genotypesort[k]=='GT'){color
           [k]<-"black"}
        else if(genotypesort[k]=='CG'){color
           [k]<-"brown"}
}
#Plot the Allele #1 vs Allele #2 signal
   clustered by genotype
plot(1,type="n",xlim=c(-0.5,2.5),ylim=c
   (-0.5,2.5),xlab='Allele␣1',ylab='Allele␣2
   ',main=as.character(ListToLook[j]))
points(allele1[1,genotype==as.character(
   genotypesort[1])],allele2[1,genotype==as.
   character(genotypesort[1])],col=color[1])
legend(2,2, c(as.character(genotypesort[1])
   ), cex=0.8, col=c(color[1]), pch=21:22,
   lty=1:2)}

if(length(unique(genotype))==2){
for(k in 1:2){
        if(genotypesort[k]=='AA'){color[k]<-
           "red"}
        else if(genotypesort[k]=='TT'){color
           [k]<-"green"}
        else if(genotypesort[k]=='CC'){color
           [k]<-"yellow"}
        else if(genotypesort[k]=='GG'){color
           [k]<-"blue"}
        else if(genotypesort[k]=='AT'){color
           [k]<-"orange"}
        else if(genotypesort[k]=='AC'){color
           [k]<-"grey"}
        else if(genotypesort[k]=='AG'){color
           [k]<-"purple"}
        else if(genotypesort[k]=='CT'){color
           [k]<-"pink"}
        else if(genotypesort[k]=='GT'){color
           [k]<-"black"}
        else if(genotypesort[k]=='CG'){color
           [k]<-"brown"}
}
#Plot the Allele #1 vs Allele #2 signal
   clustered by genotype
```

```
    plot(1,type="n",xlim=c(-0.5,2.5),ylim=c
       (-0.5,2.5),xlab='Allele␣1',ylab='Allele␣2
       ',main=as.character(ListToLook[j]))
    points(allele1[1,genotype==as.character(
       genotypesort[1])],allele2[1,genotype==as.
       character(genotypesort[1])],col=color[1])
    points(allele1[1,genotype==as.character(
       genotypesort[2])],allele2[1,genotype==as.
       character(genotypesort[2])],col=color[2])
    legend(2,2, c(as.character(genotypesort[1])
       ,as.character(genotypesort[2])), cex=0.8,
        col=c(color[1],color[2]), pch=21:22, lty
       =1:2)}

if(length(unique(genotype))==3){
    for(k in 1:3){
           if(genotypesort[k]=='AA'){color[k]<-
              "red"}
           else if(genotypesort[k]=='TT'){color
              [k]<-"green"}
           else if(genotypesort[k]=='CC'){color
              [k]<-"yellow"}
           else if(genotypesort[k]=='GG'){color
              [k]<-"blue"}
           else if(genotypesort[k]=='AT'){color
              [k]<-"orange"}
           else if(genotypesort[k]=='AC'){color
              [k]<-"grey"}
           else if(genotypesort[k]=='AG'){color
              [k]<-"purple"}
           else if(genotypesort[k]=='CT'){color
              [k]<-"pink"}
           else if(genotypesort[k]=='GT'){color
              [k]<-"black"}
           else if(genotypesort[k]=='CG'){color
              [k]<-"brown"}
            }
    plot(1,type="n",xlim=c(-0.5,2.5),ylim=c
       (-0.5,2.5),xlab='Allele␣1',ylab='Allele␣2
       ',main=as.character(ListToLook[j]))
    points(allele1[1,genotype==as.character(
       genotypesort[1])],allele2[1,genotype==as.
       character(genotypesort[1])],col=color[1])
    points(allele1[1,genotype==as.character(
       genotypesort[2])],allele2[1,genotype==as.
       character(genotypesort[2])],col=color[2])
```

```
                points(allele1[1,genotype==as.character(
                   genotypesort[3])],allele2[1,genotype==as.
                   character(genotypesort[3])],col=color[3])
                legend(2,2, c(as.character(genotypesort[1])
                   ,as.character(genotypesort[2]),as.
                   character(genotypesort[3])), cex=0.8, col
                   =c(color[1],color[2],color[3]), pch
                   =21:22, lty=1:2)}

                #Save the plot as a PDF file
                outfile=paste(as.character(ListToLook[j]),'
                   Cases.pdf',sep="")
                dev.copy(pdf,outfile)
                dev.off()
        }
}

#FOR CONTROL

for (i in 3:3){
        #Load list of SNPs to plot, first column should be
           SNP, second column chromosome location
        listToLook=read.table("ListToLook.txt",sep="\t",
           header=FALSE)
        #Get the SNPs corresponding to the ith Chromosome
        ListToLook=listToLook[listToLook[,2]==i,1]

        #Read the genotype file for the 58C group
        infile=paste("Affx_20070205fs1_gt_58C_0",i,".txt",
           sep="")
        gene=read.table(infile,header=TRUE)

        #Get the genotype for each SNP
        keep=which(as.character(gene[,1]) %in% as.character
           (ListToLook))
        genoControls58C=gene[keep,c(1,3)]

        #Remove genotype file
        rm(gene)

        #Read the genotype file for the NBS group
        infile=paste("Affx_20070205fs1_gt_NBS_0",i,".txt",
           sep="")
        gene=read.table(infile,header=TRUE)

        #Get the genotype for each SNP
```

```
keep=which(as.character(gene[,1]) %in% as.character
   (ListToLook))
genoControlsNBS=gene[keep,c(1,3)]

#Remove genotype file
rm(gene)

#Read the signal file for the 58C group
infile=paste("Affx_20070205f_signal_58C_0",i,".txt"
   ,sep="")
gene=read.table(infile,header=TRUE)

#Get the signal for each SNP
keep=which(as.character(gene[,2]) %in% as.character
   (ListToLook))
allelesControls58C=gene[keep,]

#Remove genotype file
rm(gene)

#Read the signal file for the NBS group
   infile=paste("Affx_20070205f_signal_NBS_0",i,".
      txt",sep="")
gene=read.table(infile,header=TRUE)

#Get the signal for each SNP
keep=which(as.character(gene[,2]) %in% as.character
   (ListToLook))
allelesControlsNBS=gene[keep,]

#Remove genotype file
rm(gene)

for (j in 1:length(ListToLook)){
        #Get the Allele #1 and Allele #2 for each
           SNP
        allele1=allelesControls58C[
           allelesControls58C[,2]==as.character(
           ListToLook[j]),seq(6,dim(
           allelesControls58C)[2],by=2)]
        allele2=allelesControls58C[
           allelesControls58C[,2]==as.character(
           ListToLook[j]),seq(7,dim(
           allelesControls58C)[2],by=2)]
        allele3=allelesControlsNBS[
           allelesControlsNBS[,2]==as.character(
           ListToLook[j]),seq(6,dim(
```

```
            allelesControlsNBS)[2],by=2)]
        allele4=allelesControlsNBS[
            allelesControlsNBS[,2]==as.character(
            ListToLook[j]),seq(7,dim(
            allelesControlsNBS)[2],by=2)]

        #Get the genotype for each SNP
        genotype58C=genoControls58C[genoControls58C
            [,1]==as.character(ListToLook[j]),2]
        genotypeNBS=genoControlsNBS[genoControlsNBS
            [,1]==as.character(ListToLook[j]),2]
        genotypesort=sort(unique(genotype58C))
        color<-c('color1','color2','color3')
if(length(unique(genotype58C))==1){
        for(k in 1:1){
              if(genotypesort[k]=='AA'){color[k]<-
                 "red"}
              else if(genotypesort[k]=='TT'){color
                 [k]<-"green"}
              else if(genotypesort[k]=='CC'){color
                 [k]<-"yellow"}
              else if(genotypesort[k]=='GG'){color
                 [k]<-"blue"}
              else if(genotypesort[k]=='AT'){color
                 [k]<-"orange"}
              else if(genotypesort[k]=='AC'){color
                 [k]<-"grey"}
              else if(genotypesort[k]=='AG'){color
                 [k]<-"purple"}
              else if(genotypesort[k]=='CT'){color
                 [k]<-"pink"}
              else if(genotypesort[k]=='GT'){color
                 [k]<-"black"}
              else if(genotypesort[k]=='CG'){color
                 [k]<-"brown"}

        }
        #Plot the Allele #1 vs Allele #2 signal
            clustered by genotype
        plot(1,type="n",xlim=c(-0.5,2.5),ylim=c
            (-0.5,2.5),xlab='Allele␣1',ylab='Allele␣2
            ',main=as.character(ListToLook[j]))
        points(allele1[1,genotype58C==as.character(
            genotypesort[1])],allele2[1,genotype58C==
            as.character(genotypesort[1])],col=color
            [1])
```

```
        points(allele3[1,genotypeNBS==as.character(
           genotypesort[1])],allele4[1,genotypeNBS==
           as.character(genotypesort[1])],col=color
           [1])
        legend(2,2, c(as.character(genotypesort[1])
           ), cex=0.8, col=c(color[1]), pch=21:22,
           lty=1:2)}
if(length(unique(genotype58C))==2){
        for(k in 1:2){
              if(genotypesort[k]=='AA'){color[k]<-
                 "red"}
              else if(genotypesort[k]=='TT'){color
                 [k]<-"green"}
              else if(genotypesort[k]=='CC'){color
                 [k]<-"yellow"}
              else if(genotypesort[k]=='GG'){color
                 [k]<-"blue"}
              else if(genotypesort[k]=='AT'){color
                 [k]<-"orange"}
              else if(genotypesort[k]=='AC'){color
                 [k]<-"grey"}
              else if(genotypesort[k]=='AG'){color
                 [k]<-"purple"}
              else if(genotypesort[k]=='CT'){color
                 [k]<-"pink"}
              else if(genotypesort[k]=='GT'){color
                 [k]<-"black"}
              else if(genotypesort[k]=='CG'){color
                 [k]<-"brown"}

        }
        #Plot the Allele #1 vs Allele #2 signal
           clustered by genotype
        plot(1,type="n",xlim=c(-0.5,2.5),ylim=c
           (-0.5,2.5),xlab='Allele␣1',ylab='Allele␣2
           ',main=as.character(ListToLook[j]))
        points(allele1[1,genotype58C==as.character(
           genotypesort[1])],allele2[1,genotype58C==
           as.character(genotypesort[1])],col=color
           [1])
        points(allele1[1,genotype58C==as.character(
           genotypesort[2])],allele2[1,genotype58C==
           as.character(genotypesort[2])],col=color
           [2])
        points(allele3[1,genotypeNBS==as.character(
           genotypesort[1])],allele4[1,genotypeNBS==
           as.character(genotypesort[1])],col=color
```

```
            [1])
          points(allele3[1,genotypeNBS==as.character(
             genotypesort[2])],allele4[1,genotypeNBS==
             as.character(genotypesort[2])],col=color
             [2])
          legend(2,2, c(as.character(genotypesort[1])
             ,as.character(genotypesort[2])), cex=0.8,
              col=c(color[1],color[2]), pch=21:22, lty
             =1:2)}

if(length(unique(genotype58C))==3){
     for(k in 1:3){
                if(genotypesort[k]=='AA'){color[k]<-
                   "red"}
                else if(genotypesort[k]=='TT'){color
                   [k]<-"green"}
                else if(genotypesort[k]=='CC'){color
                   [k]<-"yellow"}
                else if(genotypesort[k]=='GG'){color
                   [k]<-"blue"}
                else if(genotypesort[k]=='AT'){color
                   [k]<-"orange"}
                else if(genotypesort[k]=='AC'){color
                   [k]<-"grey"}
                else if(genotypesort[k]=='AG'){color
                   [k]<-"purple"}
                else if(genotypesort[k]=='CT'){color
                   [k]<-"pink"}
                else if(genotypesort[k]=='GT'){color
                   [k]<-"black"}
                else if(genotypesort[k]=='CG'){color
                   [k]<-"brown"}

          }
    plot(1,type="n",xlim=c(-0.5,2.5),ylim=c
       (-0.5,2.5),xlab='Allele␣1',ylab='Allele␣2',
       main=as.character(ListToLook[j]))
          points(allele1[1,genotype58C==as.character(
             genotypesort[1])],allele2[1,genotype58C==
             as.character(genotypesort[1])],col=color
             [1])
          points(allele1[1,genotype58C==as.character(
             genotypesort[2])],allele2[1,genotype58C==
             as.character(genotypesort[2])],col=color
             [2])
          points(allele1[1,genotype58C==as.character(
             genotypesort[3])],allele2[1,genotype58C==
```

```
                    as.character(genotypesort[3])],col=color
                    [3])
                points(allele3[1,genotypeNBS==as.character(
                    genotypesort[1])],allele4[1,genotypeNBS==
                    as.character(genotypesort[1])],col=color
                    [1])
                points(allele3[1,genotypeNBS==as.character(
                    genotypesort[2])],allele4[1,genotypeNBS==
                    as.character(genotypesort[2])],col=color
                    [2])
                points(allele3[1,genotypeNBS==as.character(
                    genotypesort[3])],allele4[1,genotypeNBS==
                    as.character(genotypesort[3])],col=color
                    [3])
                legend(2,2, c(as.character(genotypesort[1])
                    ,as.character(genotypesort[2]),as.
                    character(genotypesort[3])), cex=0.8, col
                    =c(color[1],color[2],color[3]), pch
                    =21:22, lty=1:2)}


                #Save the plot as a PDF file
                outfile=paste(as.character(ListToLook[j]),'
                    Controls.pdf',sep="")
                dev.copy(pdf,outfile)
                dev.off()
        }
}

########################################
######CROSS VALIDATION IN LOGIC REGRESSION######
########################################

rm(list=ls())

args=commandArgs();
##use this to divide to different jobs to parallel run ##
start=as.numeric(args[4]);
end=as.numeric(args[5]);

library(LogicReg)

# Reading phenotype data
aa=read.table("pheno.txt",header=T,sep="\t")
resp=aa$pd
length(resp)
sum(resp)
```

```
path=NULL

# Reading the top 123 genes data
path = read.table("topgenedata.txt", header=T)
print(dim(path))

dev=NULL
for (i in start:end){
        set.seed(i)
        res=logreg(resp=resp,bin=t(path[,-c(1:5)]),type=3,
           select=3,ntrees=c(1,9),nleaves=c(1,20))
        dev=res$cvscores
}

devout=paste("Top_Gene_CV_1_9","_",start,"_",end,".txt",sep
   ="")
write.table(dev,devout,col.names=F,sep="\t")

##################################
############LOGIC REGRESSION########
##################################

rm(list=ls())

args=commandArgs();
##use this to divide to different jobs to parallel run ##
start=as.numeric(args[4]);
end=as.numeric(args[5]);

library(LogicReg)

# Reading phenotype data
aa=read.table("pheno.txt",header=T,sep="\t")
resp=aa$pd
length(resp)
sum(resp)

path = read.table("topgenedata.txt", header=T)
print(dim(path))

dev=NULL
any1=NULL

for (i in start:end){
        set.seed(i)
```

```
        res=logreg(resp=resp,bin=t(path[,-c(1:5)]),type=3,
           select=1,ntrees=9,nleaves=17)
        any1=c(res$model$score, i, k)
        dev= rbind(dev, any1)
}

devout=paste("Res","_",start,"_",end,".txt",sep="")
write.table(dev,devout,col.names=F,sep="\t")

#####################################
###########NON-NESTED MODEL TEST#########
#####################################

#Reading independent SNP data
snp17 = read.table("indSNPdata.txt", header=T)

#Reading logic tree data
tree9_data = read.table("tree9dataL.txt", header=T)

m_snp17<- glm(snp17[,18] ~ snp17[,1]+snp17[,2]+snp17[,3]+
   snp17[,4]+snp17[,5]+snp17[,6]+snp17[,7]+snp17[,8]+snp17
   [,9]+snp17[,10]+snp17[,11]+snp17[,12]+snp17[,13]+snp17
   [,14]+snp17[,15]+snp17[,16]+snp17[,17], data = snp17,
   family=binomial)
m_tree9<- glm(tree9_data[,10] ~ tree9_data[,1]+tree9_data
   [,2]+tree9_data[,3]+tree9_data[,4]+tree9_data[,5]+tree9_
   data[,6]+tree9_data[,7]+tree9_data[,8]+tree9_data[,9],
   data = tree9_data, family=binomial)

library(games)
clarke(m_snp17, m_tree9)

#####################################
###CROSS-VALIDATED LOG ODDS RATIO PLOT###
#####################################

rm(list=ls())

args=commandArgs();
####use this to divide to different jobs to parallel run
   ####
start=as.numeric(args[4]);
end=as.numeric(args[5]);

yourData<- read.table("CD_data.txt", header = F)
pheno<- read.table("pheno.txt", header = T)
```

```
colnames(yourData)[1:6]<- c("sl", "snp", "chr", "position",
    "gene", "etc")
#colnames(yourData)[7:4690]<- rownames(pheno)

rownames(pheno)<- colnames(yourData)[7:4690]

#### Partitioning case and control ####
dim(yourData)
dim(pheno)

myData<- t(yourData[,-c(1:6)])
colnames(myData)<- as.character(yourData[,1])
myData_p<- cbind(myData, pheno$pd)
colnames(myData_p)[8059]<- "resp"

yourData_case<- myData_p[myData_p[,8059]==1,]
yourData_control<- myData_p[myData_p[,8059]==0,]

#### Shuffling ####
set.seed(2010)
yourData_case<-yourData_case[sample(nrow(yourData_case)),]
yourData_case[1:10, 1:10]
set.seed(2020)
yourData_control<-yourData_control[sample(nrow(yourData_
   control)),]
yourData_control[1:10, 1:10]

#Create 10 equally size folds
folds_case <- cut(seq(1,nrow(yourData_case)),breaks=10,
   labels=FALSE)
folds_control <- cut(seq(1,nrow(yourData_control)),breaks
   =10,labels=FALSE)

#Perform 10 fold cross validation

test_set<- NULL
for(i in start:end){
    #Segement your data by fold using the which() function
    testIndexes_case <- which(folds_case==i,arr.ind=TRUE)
    testIndexes_control <- which(folds_control==i,arr.ind=
       TRUE)

### Generating training and test set ###
# Case #
    testData_case <- yourData_case[testIndexes_case, ]
    trainData_case <- yourData_case[-testIndexes_case, ]
# Control #
```

```
    testData_control <- yourData_control[testIndexes_
       control, ]
    trainData_control <- yourData_control[-testIndexes_
       control, ]

## Merging the case and control training dataset ##
training_data<- rbind(trainData_case, trainData_control)
resp<- training_data[,8059]
td1<- t(training_data[,-8059])
td2<- cbind(yourData[,c(1:6)], td1)

## Merging the case and control training dataset ##
test_data<- rbind(testData_case, testData_control)

library(LogicReg)

set.seed(2010)
seed<- sample(1000:5000, 30, replace=F)

score<- NULL
for (j in 1:length(seed)){
set.seed(seed[j])
y<- logreg(resp=resp,bin=t(td2[,-c(1:6)]),type=3,select=1,
   ntrees=9,nleaves=17)
devi<- c(y$model$score, seed[j])
score<- rbind(score, devi)
}

min_score<- score[which(score[,1] %in% min(score[,1])),2]
print(min_score)

set.seed(min_score)
getY<- logreg(resp=resp,bin=t(td2[,-c(1:6)]),type=3,select
   =1,ntrees=9,nleaves=17)
print(getY)

getY$model[[5]][[1]]  ####tree 1 information
tree=as.character(getY$model[[5]][[1]][3])
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree1x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

getY$model[[5]][[2]]  ####tree 2 information
tree=as.character(getY$model[[5]][[2]][3])
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
```

```
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree2x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

getY$model[[5]][[3]]  ####tree 3 information
tree=as.character(getY$model[[5]][[3]][3])
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree3x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

getY$model[[5]][[4]]  ####tree 4 information
tree=as.character(getY$model[[5]][[4]][3])
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree4x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

getY$model[[5]][[5]]  ####tree 5 information
tree=as.character(getY$model[[5]][[5]][3])
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree5x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

getY$model[[5]][[6]]  ####tree 6 information
tree=as.character(getY$model[[5]][[6]][3])
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree6x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

getY$model[[5]][[7]]  ####tree 7 information
tree=as.character(getY$model[[5]][[7]][3])
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree7x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

getY$model[[5]][[8]]  ####tree 8 information
tree=as.character(getY$model[[5]][[8]][3])
```

```
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree8x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

getY$model[[5]][[9]]  ####tree 9 information
tree=as.character(getY$model[[5]][[9]][3])
xs1=(strsplit(tree,"knot␣=␣c"))[[1]][2]
xs2=strsplit(xs1,"),␣neg")
x3=strsplit(xs2[[1]][1],'\\(')   ####this contains the x
   variable number, if not 0
tree9x=as.numeric(strsplit(x3[[1]][2],",")[[1]])

both=c(tree1x, tree2x, tree3x, tree4x, tree5x, tree6x,
   tree7x, tree8x, tree9x)
rows=both[both!=0]

leaf<- data.frame(rows, td2[rows,1:6])
print(leaf)

OR_train<- exp(getY$model$coef[2:10])
print(OR_train)

test_set<- rbind(test_set, test_data)
}

devout=paste("Res","_",start,"_",end,".txt",sep="")
write.table(test_set,devout,col.names=T, row.names=T, sep="
   \t")

###################################
#THE FOLLOWING CODE IS REPEATED FOR THE REMAINING 9-FOLD OF
    CROSS-VALIDATION AS WELL

rm(list=ls())

#Fold 1

test_data<- read.table("Res_1_1.txt", header=T)

tree1<- test_data$X9339
tree2<- as.numeric(test_data$X5138 & (1- test_data$X2811))
tree3<- as.numeric(test_data$X22301 & (1- test_data$X12005)
   )
tree4<- test_data$X3065
```

```
tree5<- as.numeric((1- test_data$X14253) & (1- test_data$
   X15307))
tree6<- as.numeric(test_data$X8181 & (1- test_data$X17684))
tree7<- as.numeric((1- test_data$X2877) | test_data$X16029)
tree8<- as.numeric(test_data$X588 | (1- test_data$X28009))
tree9<- as.numeric(((1- test_data$X24507) | test_data$
   X11980) & (1- test_data$X2179 ))

val_test<- data.frame(tree1, tree2, tree3, tree4, tree5,
   tree6, tree7, tree8, tree9, resp=test_data[,8059])

OR_train<- exp(c(-0.349,   +0.445,  -0.521,   +0.477,
   +0.486,   +0.368,   -0.439,  +4.17,   -0.622))

val_test$or_t1<- rep(1, nrow(val_test))
val_test$or_t1[val_test$tree1 == 1]<- OR_train[1]
val_test$or_t2<- rep(1, nrow(val_test))
val_test$or_t2[val_test$tree2 == 1]<- OR_train[2]
val_test$or_t3<- rep(1, nrow(val_test))
val_test$or_t3[val_test$tree3 == 1]<- OR_train[3]
val_test$or_t4<- rep(1, nrow(val_test))
val_test$or_t4[val_test$tree4 == 1]<- OR_train[4]
val_test$or_t5<- rep(1, nrow(val_test))
val_test$or_t5[val_test$tree5 == 1]<- OR_train[5]
val_test$or_t6<- rep(1, nrow(val_test))
val_test$or_t6[val_test$tree6 == 1]<- OR_train[6]
val_test$or_t7<- rep(1, nrow(val_test))
val_test$or_t7[val_test$tree7 == 1]<- OR_train[7]
val_test$or_t8<- rep(1, nrow(val_test))
val_test$or_t8[val_test$tree8 == 1]<- OR_train[8]
val_test$or_t9<- rep(1, nrow(val_test))
val_test$or_t9[val_test$tree9 == 1]<- OR_train[9]

val_test$OR_total<- val_test$or_t1 * val_test$or_t2 * val_
   test$or_t3* val_test$or_t4* val_test$or_t5 * val_test$or_
   t6 * val_test$or_t7* val_test$or_t8* val_test$or_t9

rownames(val_test)<- rownames(test_data)

write.csv(val_test, "Test1.csv", row.names=T)

##########################################

rm(list=ls())

test1<- read.csv("Test1.csv", header=T)
test2<- read.csv("Test2.csv", header=T)
```

```
test3<- read.csv("Test3.csv", header=T)
test4<- read.csv("Test4.csv", header=T)
test5<- read.csv("Test5.csv", header=T)
test6<- read.csv("Test6.csv", header=T)
test7<- read.csv("Test7.csv", header=T)
test8<- read.csv("Test8.csv", header=T)
test9<- read.csv("Test9.csv", header=T)
test10<- read.csv("Test10.csv", header=T)

test_data<- rbind(test1, test2, test3, test4, test5, test6,
    test7, test8, test9, test10)
write.csv(test_data, "test_data.csv", row.names=T)

or_data<- test_data

#now make your lovely plot
library(ggplot2)

pdf("CV_OR_plot_CD.pdf", height=5, width=7)

RR_case<- log(or_data$OR_total[or_data$resp==1])
RR_cont<- log(or_data$OR_total[or_data$resp==0])

## calculate the density - don't plot yet
densCase <- density(RR_case)
densControl <- density(RR_cont)
## calculate the range of the graph
xlim <- range(densControl$x,densCase$x)
ylim <- range(0,densControl$y, densCase$y)
#pick the colours
caseCol <- rgb(1,0,0,0.4)
contCol <- rgb(0,0,1,0.4)
## plot the carrots and set up most of the plot parameters
plot(densCase, xlim = xlim, ylim = ylim, xlab = 'log Total
   Odds Ratio',
     main = 'Distribution of log Total Odds Ratio \nfor
        Cases and Controls',
     panel.first = grid())
#put our density plots in
polygon(densCase, density = -1, col = caseCol)
polygon(densControl, density = -1, col = contCol)
## add a legend in the corner
legend('topleft',c('Cases','Controls'),
       fill = c(caseCol, contCol), bty = 'n',
       border = NA)

dev.off()
```

A.2 Two Stage Procedure to Check Interaction across Whole Genome

```
##############################################
###FILTERING BY MARGINAL ASSOCIATION IN DATASET 1###
##############################################

rm(list=ls(all=TRUE))

args=commandArgs();
##use this to divide to different jobs to parallel run ##
start=as.numeric(args[4]);
end=as.numeric(args[5]);

library(lmtest)

for (chr in start:end){

genodata<- read.table("A01Chr1.txt", header=T)
pheno<- read.table("pheno.txt", header=T)
res<- pheno$pd

dim(genodata)

odds<- seq(1,dim(genodata)[1],2)

me_all<- NULL
for (i in odds){
        me<- glm(res~t(genodata[i,-c(1:4)]) + t(genodata[i
           +1,-c(1:4)]), family=binomial)
        or1<- exp(me$coefficients[2])
        or2<- exp(me$coefficients[3])
        pval1<- summary(me)$coef[,"Pr(>|z|)"][2]
        pval2<- summary(me)$coef[,"Pr(>|z|)"][3]
        menull<- glm(res~1, family=binomial)
        lrt<- lrtest(me, menull)$"Pr(>Chisq)"[2]
        me_sum<- data.frame(genodata[i,1], genodata[i,2],
           genodata[i,4], or1, or2, pval1, pval2, lrt)
        me_all<- rbind(me_all, me_sum)
}
}

dim(me_all)

devout=paste("Me_Chr1","_",start,"_",end,".txt",sep="")
```

```
write.table(me_all,devout,col.names=F,row.names=F, sep="\t"
   )

### THE PROCESS IS REPEATED FOR ALL 22 CHROMOSOMES OF
   DATASET 1.

##########################################
##CROSS VALIDATION FOR LOGIC REGRESSION FOR DATASET 1##
##########################################

rm(list=ls())

args=commandArgs();
####use this to divide to different jobs to parallel run
   ####
start=as.numeric(args[4]);
end=as.numeric(args[5]);

library(LogicReg)

# Reading phenotype data
aa=read.table("pheno.txt",header=T,sep="\t")
resp=aa$pd
length(resp)
sum(resp)

# Reading filtered SNPs that are marginally significant.
path = read.table("Marginal_Sig_SNP_data.txt", header=T)
print(dim(path))

set.seed(2014)
ran_seed<- sample(1000:5000, 30, replace=FALSE)

dev=NULL
for (i in start:end){
        set.seed(ran_seed[i])
        res=logreg(resp=resp,bin=t(path[,-c(1:4)]),type=3,
           select=3,ntrees=c(1,9),nleaves=c(1,25))
        dev=res$cvscores
}

devout=paste("CV","_",start,"_",end,".txt",sep="")
write.table(dev,devout,col.names=F,sep="\t")

############################
###LOGIC REGRESSION IN DATASET 1###
############################
```

```
rm(list=ls())

args=commandArgs();
####use this to divide to different jobs to parallel run
   ####
start=as.numeric(args[4]);
end=as.numeric(args[5]);

library(LogicReg)

# Reading outcome data
aa=read.table("pheno.txt",header=T,sep="\t")
resp=aa$pd
length(resp)
sum(resp)

path = read.table("Marginal_Sig_SNP_data.txt", header=T)
print(dim(path))

set.seed(2100)
ran_seed<- sample(2900:3000, 30, replace=FALSE)

dev=NULL
any1=NULL

for (i in start:end){
        set.seed(ran_seed[i])
        res=logreg(resp=resp,bin=t(path[,-c(1:4)]),type=3,
           select=1,ntrees=5,nleaves=19)
        any1=cbind(res$model$score, ran_seed[i])
        dev= rbind(dev, any1)
}

devout=paste("Res","_",start,"_",end,".txt",sep="")
write.table(dev,devout,col.names=F, row.names=F, sep="\t")

#############################################################
#################NON-NESTED MODEL TEST#####################
#############################################################

#Reading independent SNP data

indSNP = read.table("indSNPdata.txt", header=T)
snp19 = indSNP

#Reading logic tree data
```

```
tree5_data = read.table("tree5dataL.txt", header=T)

m_snp19<- glm(snp19[,20] ~ snp19[,1]+snp19[,2]+snp19[,3]+
   snp19[,4]+snp19[,5]+snp19[,6]+snp19[,7]+snp19[,8]+snp19
   [,9]+snp19[,10]+snp19[,11]+snp19[,12]+snp19[,13]+snp19
   [,14]+snp19[,15]+snp19[,16]+snp19[,17]+snp19[,18]+snp19
   [,19], data = snp19, family=binomial)
m_tree5<- glm(tree5_data[,6] ~ tree5_data[,1]+tree5_data
   [,2]+tree5_data[,3]+tree5_data[,4]+tree5_data[,5], data =
    tree5_data, family=binomial)

library(games)

clarke(m_snp19, m_tree5)
```

A.3 Exploration of the SNP Pairs

```
#################################################
###SEARCH FOR SIGNIFICANT SNP PAIRS IN CASES BUT NOT IN
   CONTROL###
#################################################

rm(list=ls(all=TRUE))

args=commandArgs();
####use this to divide to different jobs to parallel run
   ####
start=as.numeric(args[4]);
end=as.numeric(args[5]);

filtered<- read.table("Filtered_Data.txt", header=F)
pheno<- read.table("pheno.txt", header=T)
res<- pheno$pd

me_all<- NULL
for (i in start:end){
fil_loop<- filtered[-i,]
        for (j in 1:dim(fil_loop)[1]){
                pair_search<- table(t(filtered[i, -c(1:5)])
                   , t(fil_loop[j, -c(1:5)]), res)[,,2]>=13
                   & table(t(filtered[i, -c(1:5)]), t(fil_
                   loop[j, -c(1:5)]), res)[,,1]==0
                geno_pos<- which(pair_search=="TRUE")
                gen<- paste(geno_pos, collapse=',')
```

```
                int_sum<- data.frame(filtered[i,1],
                   filtered[i,2], filtered[i,3], filtered[i
                   ,5], fil_loop[j,1], fil_loop[j,2], fil_
                   loop[j,3], fil_loop[j,5], gen)
                me_all<- rbind(me_all, int_sum)
        }
}

dim(me_all)

devout=paste("X-Z_Int","_",start,"_",end,".txt",sep="")
write.table(me_all,devout,col.names=F,row.names=F, sep="\t"
   )

#
   ##############################################################

#GENERATING DATA WITH PATTERNS OF POSITIVES IN SIGNIFICANT
   SNP PAIRS IN DATASET 1 CASES#
#
   ##############################################################


rm(list=ls(all=TRUE))

args=commandArgs();
####use this to divide to different jobs to parallel run
   ####
start=as.numeric(args[4]);
end=as.numeric(args[5]);

# Reading the SNP data which has been filtered based on
   marginal association
filtered<- read.table("Filtered_Data.txt", header=F)
phenotype<- read.table("pheno.txt", header=T)
res<- phenotype$pd

colnames(filtered)[6:1225]<- as.character(phenotype$id)

highriskpair<- read.csv("Pair_Analysis2x3.csv", header=T)

seq_num<- data.frame(start = seq(6,1225, 20), end = seq(25,
    1225, 20))

samples=NULL
for (j in seq_num[start,1]:seq_num[end,2]){
riskpair=NULL
```

```
        for (i in 1:nrow(highriskpair)){
                x<- sum(filtered[,j][filtered[,2] %in%
                   highriskpair[i,1] & filtered[,5] %in%
                   highriskpair[i,3]]==highriskpair[i,7] &
                   filtered[,j][filtered[,2] %in%
                   highriskpair[i,4] & filtered[,5] %in%
                   highriskpair[i,6]]==highriskpair[i,8])
                riskpair<- cbind(riskpair,x)
        }
        samples<- rbind(samples,riskpair)
}

dim(samples)

devout=paste("Riskpair_percase","_",start,"_",end,".txt",
   sep="")
write.table(samples,devout,col.names=F,row.names=F, sep="\t
   ")

############################
##MAPPING DATASET 1 AND DATASET 2##
############################

rm(list=ls(all=TRUE))

filtered<- read.table("FilteredRawD1.txt", header=F)

gene<- as.matrix(filtered[,-c(1:3)])

#####MAKE THE GENOTYPES UNIFORM#####
gene[(gene=='0')]='00'
gene[(gene=='TA')]='AT'
gene[(gene=='CA')]='AC'
gene[(gene=='GA')]='AG'
gene[(gene=='TC')]='CT'
gene[(gene=='TG')]='GT'
gene[(gene=='GC')]='CG'

devout=paste("FilteredRawThaiUniform.txt",sep="")
write.table(gene,devout,col.names=F,row.names=F, sep="\t")

highriskpair<- read.csv("Pair_Analysis2x3.csv", header=T)

JJThai_filtered<- read.table("JJsnpThaiFiltered.txt",
   header=F)
JJThai_filtered<- as.matrix(JJThai_filtered[,-1])
```

```
ThaiMappedJJ=highriskpair[which((highriskpair[,1] %in%
   JJThai_filtered[,1]) & (highriskpair[,4] %in% JJThai_
   filtered[,1])),]

UniformThaiGeno<- read.table("FinalGenoThai.txt", header=F)
UniformThaiGeno<- as.matrix(UniformThaiGeno[,-1])

ThaiMappedJJ$Rawgtsnp1<- rep("NA", nrow(ThaiMappedJJ))
ThaiMappedJJ$Rawgtsnp2<- rep("NA", nrow(ThaiMappedJJ))

for (i in 1:nrow(ThaiMappedJJ)){
        if (ThaiMappedJJ[i,7]==2){
                        ThaiMappedJJ[i,9]=UniformThaiGeno[
                           UniformThaiGeno[,1]==ThaiMappedJJ
                           [i,1],2]
        } else {
        if (ThaiMappedJJ[i,7]==1){
                        ThaiMappedJJ[i,9]=UniformThaiGeno[
                           UniformThaiGeno[,1]==ThaiMappedJJ
                           [i,1],3]
        } else {
        if (ThaiMappedJJ[i,7]==0){
                        ThaiMappedJJ[i,9]=UniformThaiGeno[
                           UniformThaiGeno[,1]==ThaiMappedJJ
                           [i,1],4]
        }
        }
        }
}

for (i in 1:nrow(ThaiMappedJJ)){
        if (ThaiMappedJJ[i,8]==2){
                        ThaiMappedJJ[i,10]=UniformThaiGeno[
                           UniformThaiGeno[,1]==ThaiMappedJJ
                           [i,4],2]
        } else {
        if (ThaiMappedJJ[i,8]==1){
                        ThaiMappedJJ[i,10]=UniformThaiGeno[
                           UniformThaiGeno[,1]==ThaiMappedJJ
                           [i,4],3]
        } else {
        if (ThaiMappedJJ[i,8]==0){
                        ThaiMappedJJ[i,10]=UniformThaiGeno[
                           UniformThaiGeno[,1]==ThaiMappedJJ
                           [i,4],4]
        }
        }
```

```
        }
}

TMJ=paste("Risk_Pairs_Thai.txt",sep="")
write.table(ThaiMappedJJ,TMJ,col.names=F,row.names=F, sep="
   \t")

##############################
#####FACTOR ANALYSIS IN DATASET 1#####
##############################

rm(list=ls(all=TRUE))

JJpairs<- read.table("Uncor_data_JJ_Thai_Mapped.txt",
   header=T)

# Determine Number of Factors to Extract
library(nFactors)
ev <- eigen(cor(JJpairs)) # get eigenvalues
eigenv<- ev$values

devout=paste("Eigen_val.txt",sep="")
write.table(eigenv,devout,col.names=T, row.names=T, sep="\t
   ")

# Varimax Rotated Principal Components
# Retaining 13 components
library(psych)
library(GPArotation)
fit <- principal(JJpairs, nfactors=13, rotate="varimax",
   score=TRUE)
fit # print results

loads<- as.matrix(fit$loadings)

devout=paste("Factor_loadings.txt",sep="")
write.table(loads,devout,col.names=T, row.names=T, sep="\t"
   )

loads0<- loads
for (i in 1:636){
        for (j in 1:13){
                if (loads0[i,j]>=-0.4 & loads0[i,j]<=0.4){
                        loads0[i,j] = 0
                }
        }
}
```

```
devout=paste("Factor_loadings_0_0.4.txt",sep="")
write.table(loads0,devout,col.names=T, row.names=T, sep="\t
   ")

loads0.4<- read.table("Factor_loadings_0_0.4.txt", header=T
   )
loads4<- loads0.4

rc1<- which(loads4[,1]!=0)
rc2<- which(loads4[,2]!=0)
rc3<- which(loads4[,3]!=0)
rc4<- which(loads4[,4]!=0)
rc5<- which(loads4[,5]!=0)
rc6<- which(loads4[,6]!=0)
rc7<- which(loads4[,7]!=0)
rc8<- which(loads4[,8]!=0)
rc9<- which(loads4[,9]!=0)
rc10<- which(loads4[,10]!=0)
rc11<- which(loads4[,11]!=0)
rc12<- which(loads4[,12]!=0)
rc13<- which(loads4[,13]!=0)

save.image("Fact13.RData")
```

Bibliography

1. Genome-Wide Association Studies Fact Sheet. National Human Genome Research Institute, `http://www.genome.gov/20019523` (2014). [Online; accessed 29-June-2014].

2. Frazer, K. A., Ballinger, D. G., Cox, D. R., Hinds, D. A., Stuve, L. L., Gibbs, R. A., Belmont, J. W., Boudreau, A., Hardenbol, P., Leal, S. M. *et al.* A second generation human haplotype map of over 3.1 million SNPs. *Nature* **449**, 851–861 (2007).

3. Gibbs, R. A., Belmont, J. W., Hardenbol, P., Willis, T. D., Yu, F., Yang, H., Ch'ang, L.-Y., Huang, W., Liu, B., Shen, Y. *et al.* The international HapMap project. *Nature* **426**, 789–796 (2003).

4. King, R. A., Rotter, J. I., Motulsky, A. G. *et al. The genetic basis of common diseases.* (Oxford University Press, 2002).

5. Visscher, P. M., Brown, M. A., McCarthy, M. I. & Yang, J. Five years of GWAS discovery. *The American Journal of Human Genetics* **90**, 7–24 (2012).

6. HuGe Navigator Statistics 2014. `http://64.29.163.162:8080/HuGENavigator/`

gWAHit.do?query=&typeSubmitAll=+Statistics+&Mysubmit=simple&geneOrderType=geneA (2014). [Online; accessed 24-June-2014].

7. Clayton, D. G., Walker, N. M., Smyth, D. J., Pask, R., Cooper, J. D., Maier, L. M., Smink, L. J., Lam, A. C., Ovington, N. R., Stevens, H. E. *et al.* Population structure, differential bias and genomic control in a large-scale, case-control association study. *Nature genetics* **37**, 1243–1246 (2005).

8. Pearson, T. A. & Manolio, T. A. How to interpret a genome-wide association study. *Jama* **299**, 1335–1344 (2008).

9. Klein, R. J., Zeiss, C., Chew, E. Y., Tsai, J.-Y., Sackler, R. S., Haynes, C., Henning, A. K., SanGiovanni, J. P., Mane, S. M., Mayne, S. T. *et al.* Complement factor H polymorphism in age-related macular degeneration. *Science* **308**, 385–389 (2005).

10. DeWan, A., Liu, M., Hartman, S., Zhang, S. S.-M., Liu, D. T., Zhao, C., Tam, P. O., Chan, W. M., Lam, D. S., Snyder, M. *et al.* HTRA1 promoter polymorphism in wet age-related macular degeneration. *Science* **314**, 989–992 (2006).

11. Burton, P. R., Clayton, D. G., Cardon, L. R., Craddock, N., Deloukas, P., Duncanson, A., Kwiatkowski, D. P., McCarthy, M. I., Ouwehand, W. H., Samani, N. J. *et al.* Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* **447**, 661–678 (2007).

12. Franke, A., McGovern, D. P., Barrett, J. C., Wang, K., Radford-Smith, G. L., Ahmad, T., Lees, C. W., Balschun, T., Lee, J., Roberts, R. *et al.* Genome-wide meta-analysis increases to 71 the number of confirmed Crohn's disease susceptibility loci. *Nature genetics* **42**, 1118–1125 (2010).

13. Anderson, C. A., Boucher, G., Lees, C. W., Franke, A., D'Amato, M., Taylor, K. D., Lee, J. C., Goyette, P., Imielinski, M., Latiano, A. *et al.* Meta-analysis identifies 29 additional ulcerative colitis risk loci, increasing the number of confirmed associations to 47. *Nature genetics* **43**, 246–252 (2011).

14. Danoy, P., Pryce, K., Hadler, J., Bradbury, L. A., Farrar, C., Pointon, J., Ward, M., Weisman, M., Reveille, J. D., Wordsworth, B. P. *et al.* Association of variants at 1q32 and STAT3 with ankylosing spondylitis suggests genetic overlap with Crohn's disease. *PLoS genetics* **6**, e1001195 (2010).

15. Pritchard, J. K. Are rare variants responsible for susceptibility to complex diseases? *The American Journal of Human Genetics* **69**, 124–137 (2001).

16. Tabor, H. K., Risch, N. J. & Myers, R. M. Candidate-gene approaches for studying complex genetic traits: practical considerations. *Nature Reviews Genetics* **3**, 391–397 (2002).

17. Gauderman, W. J., Witte, J. S. & Thomas, D. C. Family-based association studies. *JNCI Monographs* **1999**, 31–37 (1999).

18. Laird, N. M. & Lange, C. Family-based designs in the age of large-scale gene-association studies. *Nature Reviews Genetics* **7**, 385–394 (2006).

19. Ott, J., Kamatani, Y. & Lathrop, M. Family-based designs for genome-wide association studies. *Nature Reviews Genetics* **12**, 465–474 (2011).

20. Yasui, Y. Why odds ratio estimates of GWAS are almost always close to 1.0 (2012).

21. Manolio, T. A., Collins, F. S., Cox, N. J., Goldstein, D. B., Hindorff, L. A., Hunter, D. J., McCarthy, M. I., Ramos, E. M., Cardon, L. R., Chakravarti, A. *et al.* Finding

the missing heritability of complex diseases. *Nature* **461**, 747–753 (2009).

22. Goldstein, D. B. Common genetic variation and human traits. *New England Journal of Medicine* **360**, 1696 (2009).

23. Hunter, D. J. & Kraft, P. Drinking from the fire hose—statistical issues in genomewide association studies. *N Engl J Med* **357**, 436–439 (2007).

24. Moore, J. H., Asselbergs, F. W. & Williams, S. M. Bioinformatics challenges for genome-wide association studies. *Bioinformatics* **26**, 445–455 (2010).

25. Ruczinski, I., Kooperberg, C. & LeBlanc, M. Logic regression. *Journal of Computational and Graphical Statistics* **12**, 475–511 (2003).

26. Dai, J. Y., Kooperberg, C., Leblanc, M. & Prentice, R. L. Two-stage testing procedures with independent filtering for genome-wide gene-environment interaction. *Biometrika* ass044 (2012).

27. Dinu, I., Mahasirimongkol, S., Liu, Q., Yanai, H., Eldin, N. S., Kreiter, E., Wu, X., Jabbari, S., Tokunaga, K. & Yasui, Y. SNP-SNP interactions discovered by logic regression explain Crohn's disease genetics. *PloS one* **7**, e43035 (2012).

28. Suehiro, Y., Wong, C. W., Chirieac, L. R., Kondo, Y., Shen, L., Webb, C. R., Chan, Y. W., Chan, A. S., Chan, T. L., Wu, T.-T. *et al.* Epigenetic-genetic interactions in the APC/WNT, RAS/RAF, and P53 pathways in colorectal carcinoma. *Clinical Cancer Research* **14**, 2560–2569 (2008).

29. Justenhoven, C., Hamann, U., Schubert, F., Zapatka, M., Pierl, C. B., Rabstein, S., Selinski, S., Mueller, T., Ickstadt, K., Gilbert, M. *et al.* Breast cancer: a candidate gene approach across the estrogen metabolic pathway. *Breast cancer research and*

treatment **108**, 137–149 (2008).

30. Ruczinski, C. K. I., LeBlanc, M. L. & Hsu, L. Sequence analysis using logic regression. *Genetic Epidemiology* **21**, S626–S631 (2001).

31. Kooperberg, C. & LeBlanc, M. Increasing the power of identifying gene× gene interactions in genome-wide association studies. *Genetic epidemiology* **32**, 255–263 (2008).

32. Murcray, C. E., Lewinger, J. P. & Gauderman, W. J. Gene-environment interaction in genome-wide association studies. *American journal of epidemiology* **169**, 219–226 (2009).

33. Millstein, J. Screening-testing approaches for gene-gene and gene-environment interactions using independent statistics. *Frontiers in genetics* **4** (2013).

34. Nelson, M., Bacanu, S., Mosteller, M., Li, L., Bowman, C., Roses, A., Lai, E. & Ehm, M. Genome-wide approaches to identify pharmacogenetic contributions to adverse drug reactions. *The pharmacogenomics journal* **9**, 23–33 (2008).

35. Clarke, K. A. A simple distribution-free test for nonnested model selection. *Political Analysis* **15**, 347–363 (2007).

36. Clarke, K. A. Testing nonnested models of international relations: Reevaluating realism. *American Journal of Political Science* 724–744 (2001).