UNDERSTANDING HETEROGENEITY AND INTERACTION IN THE CONTEXT OF WHOLE GENOME GENETIC ANALYSIS

by

Sriram Balu

M.D., Kharkov State Medical University, 1996

Submitted to the Graduate Faculty of

Department of Human Genetics

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Master of Science

University of Pittsburgh

2007

UNIVERSITY OF PITTSBURGH

Graduate School of Public Health

This thesis was presented

by

Sriram Balu

It was defended on

January 30, 2007

and approved by

Thesis Advisor:
Dr. Michael M. Barmada, Ph.D.
Associate Professor, Department of Human Genetics
Graduate School of Public Health
University of Pittsburgh

Committee Member:
Dr. Eleanor Feingold, Ph.D.
Associate Professor, Department of Human Genetics
Graduate School of Public Health
University of Pittsburgh

Committee Member:
Dr. John W. Wilson, Ph.D.
Assistant Professor, Department of Biostatistics
Graduate School of Public Health
University of Pittsburgh

Michael M. Barmada, Ph.D.

UNDERSTANDING HETEROGENEITY AND INTERACTION IN THE CONTEXT OF WHOLE GENOME GENETIC ANALYSIS

Sriram Balu, M.S.

University of Pittsburgh, 2007

ABSTRACT

Interactions and heterogeneity play a vital role in the miscommunication between genotype and phenotype in complex diseases. Detection of genes that influence the risk of common, complex disorders involves many statistical and computational challenges. This led us to investigate and compare the common methods of linkage analysis in complex diseases. We applied various methods of linkage analysis on the simulated dataset from the Genetic Analysis Workshop (GAW) 14. As the disease modeled in this dataset resembled a qualitative disorder, we employed methods such as nonparametric linkage scans, association studies of susceptibility regions, and conditional studies (conditioning on a previously identified susceptibility locus).

The goal of this project was to study the efficiencies and inadequacies of various methods in detecting interactions and heterogeneity in the simulated dataset.

The methods used on this dataset showed very low percentage in the detection of interactions. We attribute this unsatisfactory performance of these methods mostly to the low

prevalence of interactions in the imaginary populations studied. We also propose various ways of improving the power in these analyses like considering haplotype studies instead of targeting single markers and increasing the range of the flanking markers around regions of high LOD scores.

Public Health Importance: Understanding the complexities involved in the genetics of diseases will provide new insight for disease prevention and health promotion. For over twenty years, public health agencies have focused more and more on newborn screening programs to detect and prevent rare genetic disorders. But common complex disorders pose a bigger problem because of their unique characteristics like heterogeneity, gene-gene interactions, multiple susceptible loci, incomplete penetrance, phenocopy and presence of environmental risk factors. By comparing common methods of linkage analysis in complex disorders in the simulated dataset of Genetic Analysis Workshop (GAW) 14, our study aims to come up with a better understanding of how heterogeneity and interaction work in the context of a whole genome genetic analysis. It is also expected to lay a foundation on which future public health researchers will be able to expand on our work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

It is my primary duty to express my indebtedness to my guide and mentor, Dr. Michael Barmada, for the unwavering support I have been receiving from him right from the day I started my work under him. Sparing indeed with his time and erudition, yet, he was unsparing in his demand for excellence and perfection in the research I had taken up. Personifying ‘caring and sharing’ in the true sense of the term, Dr. Michael Barmada understood my problems and offered solutions forthwith.

I owe an ocean of gratitude to Dr. Robert Ferrell, my academic advisor, whose professional knowledge can be matched only by his own. An epitome of meticulousness and commitment, his suggestions and academic insights have gone a long way in shaping my research into its present form.

I cannot adequately thank the committee members, Dr. Daniel Weeks, Dr. Eleanor Feingold, Dr. Candace Kammerer, Dr. David Finegold and Dr. John Wilson, for the readiness and willingness with which they helped me out whenever I approached them. I thank them individually for all the timely tips and for all the succor and support I had received from them.

I am grateful to Dr. Susanne Gollin, Dr. Urvashi Surti and Dr. Sofia Shekhter-Levin for the motivation I had received from them and for the kindness shown to me from time to time.

My thanks are due to Michaele Armstrong, my batch mate, for the social, moral and academic support she had been offering me with an ever-smiling face.

I thank every member of my family in general and Chitra Chauhan in particular for being patiently behind me in my present endeavor and for generously pardoning me for my having overburdening them.

I thank one and all with whom I have been involved in this project.

LIST OF ABBREVIATIONS

chr: chromosome

htSNP: haplotype tag single nucleotide polymorphism

jck: juvenile cystic kidney

p-value: probability value

snp: single nucleotide polymorphism

AFBAC: affected family-based controls

AIC: Akaike's information criterion

APM: affected pedigree member

APOE: apolipoprotein E

ASP: affected sib-pair

BRCA1: breast cancer gene 1

BRCA2: breast cancer gene 2

CARD: caspase recruitment domain

CD: Crohn's disease

CYP: cytochrome P-450 enzyme

DNA: deoxyribonucleic acid

GAW: genetic analysis workshop

GLM: generalized linear model

HRR: haplotype relative risk

IBD: identical-by-descent

IBD: inflammatory bowel disease

IBS: identical-by-state

LD: linkage disequilibrium

LOD: logarithm of odds

LRS: likelihood ratio statistic

NIDDM: non-insulin-dependent diabetes mellitus

NOD: nucleotide-binding oligomerization domain

NPL: non-parametric linkage

OMIM: online Mendelian inheritance in man

PDT: pedigree disequilibrium test

PKD: polycystic kidney disease

PKU: phenylketonuria

TDT: transmission disequilibrium test

1.0 INTRODUCTION

The development of any complex disease is an active process that is influenced by a system of genes as well as by environmental factors. It is a statistical and computational challenge to identify and characterize genes that influence the risk of common, complex multifactorial disease. The presence of heterogeneity and interactions with other genes and environmental factors makes the task even harder.

Genetic linkage mapping is a systematic, genome-wide approach to the study of complex diseases. Genetic linkage studies of complex diseases have certainly identified susceptible chromosomal regions and provide a reliable basis for additional linkage and association studies. At least a few loci are supported by several linkage or association studies.

We set out to study common methods of linkage analysis in complex disorders in the simulated data set of Genetic Analysis Workshop (GAW) 14. As this set of simulated data mimicked a qualitative disorder, we used methods which are employed in other complex qualitative disorders, such as nonparametric linkage scans, association studies of susceptibility regions, and conditional studies (conditioning on a previously identified susceptibility locus).

2.0 OBJECTIVES

The primary objective of this study is to come up with a better understanding of how heterogeneity and interaction look from the point of view of a whole genome scan – is the evidence of linkage inflated at both loci or only one? Does conditioning on one locus find the other one if there is an interaction or only when there is heterogeneity?

To achieve these objectives we propose the following aims:

1. to devise an effective scheme to study methods of linkage analysis in the simulated data set of GAW14
2. to perform conditional analyses by applying stratification methods
3. to perform conditional analyses by applying redefinition methods
4. to perform conditional analyses by applying weighted analyses
5. to perform conditional analyses by applying logistic regression methods

3.0 LITERATURE REVIEW

3.1 SIMPLE APPROACH TO COMPLEX TRAITS

Traits that are influenced by multiple loci (genes) are called complex traits. There is no universal definition of complex trait in the literature. In a broad sense, it refers to any phenotype that does not display classic Mendelian recessive or dominant inheritance characteristic of a single gene locus.[50] However, compartmentalizing genetic disorders into simple/monogenic/Mendelian and complex/multifactorial/non-Mendelian might be an oversimplification.[17, 79] Many diseases that were initially thought to be monogenic are turning out to be either caused or modulated by the action of a small number of loci. Even the most classic 'monogenic disorders', like Phenylketonuria (PKU)[79] and sickle cell anemia[50], are now being considered as part of the spectrum of complex disorders. These disorders are described as 'oligogenic' disorders[3] – a wide range of phenotypes that are neither monogenic nor complex. Nonetheless, the etiology of most of these disorders is due to the presence of a major gene of effect with a clear inheritance pattern thereby making it relatively much easier to identify the primary genetic cause. This is evident in the OMIM database (http://www.ncbi.nlm.nih.gov/Omim/) with entry of ~2000 human diseases or traits with known molecular basis, a fairly large proportion of which are Mendelian disorders. The success attained

for the Mendelian disorders could be quite inspiring for gene-mapping efforts for complex disorders. However, there are several characteristics of the complex disorders that provide daunting challenge to such efforts.

3.1.1 Incomplete Penetrance and Phenocopy

Incomplete penetrance implies that individuals inheriting a predisposition allele may not always manifest the disease. Thus a given genotype may affect the probability of the disease but may not completely determine the outcome. Incomplete penetrance manifests itself as the disease occasionally skipping generations (Figure 1).

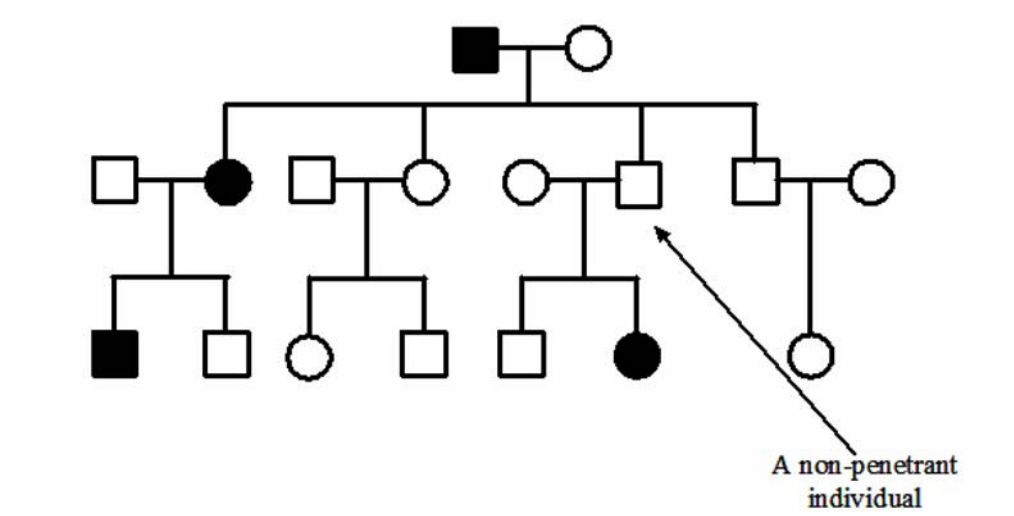

Figure 1. Incomplete penetrance. A disease skipping generations is a hallmark of incomplete penetrance

Mutations in *BRCA1* and *BRCA2* genes in the case of rare forms of breast cancer, and in *APOE* gene in the case of Alzheimer's disease are examples of incomplete penetrance.

Incomplete penetrance may be caused by interaction with other loci or environment or it could be purely a chance event. On the other hand, there may be some individuals who are affected with the same disorder but due to purely non-genetic (e.g. environmental) reasons. Such individuals (phenocopies) are clinically indistinguishable from individuals harboring the predisposition allele. Both of these phenomena hamper the gene-mapping process as the predisposition allele may be present in some unaffected individuals (incomplete penetrance) or absent in some affected individuals (phenocopy).

3.1.2 Genetic Heterogeneity

Genetic heterogeneity is a phenomenon whereby distinct mutations at the same locus (allelic heterogeneity) or at different loci (non-allelic heterogeneity) can cause the same indistinguishable phenotype. It is the non-allelic heterogeneity which is the hallmark of complex disorders. This is generally the case when different genes in the same biochemical pathway harbor the predisposition allele. For example recessive retinitis pigmentosa can be caused by at least 40 genes[95], whereas congenital hearing loss can be observed to be the result of mutations in at least 70 independent loci[88]. Such a condition poses problems to medical geneticists who can not distinguish between patients suffering from the same disease for different genetic reasons. Non-allelic heterogeneity hampers genetic mapping efforts in many ways. It often results in a single gene accounting for a small proportion of segregating families. In such situations very large families are required to obtain robust linkage evidence. Moreover, in the case of common complex disorders, in a large pedigree, chances are high that individuals carrying different predisposition genes marry into the pedigree. Then the investigator may in fact be looking at linkage of two more loci at the same time, making the linkage analysis much more difficult.[68]

Since different chromosomal regions may be involved with the disease in different families, it has been one of the major reasons for non-replication of initial gene-mapping findings' localization and conflicting position estimates of disease loci.[81]

3.1.3 Polygenic inheritance

In the case of many complex disorders, genetic heterogeneity and polygenic inheritance go together, making the gene-mapping even more complicated. Polygenic inheritance implies that multiple genetic variants, within the same or different genes, combine to affect liability for many common diseases. The variants may interact among themselves and with environmental factors (gene-gene and gene-environment interactions respectively). Polygenic traits may be classified as discrete traits or quantitative traits. Discrete traits may represent a threshold effect, produced whenever an underlying quantitative variable, influenced by multiple genes, exceeds a critical threshold, or a pure synthetic effect, requiring the simultaneous and joint action of each of several mutations.[50] Polygenic inheritance obscures genetic mapping efforts because no single locus is individually necessary and sufficient to produce a discrete trait or a high value of a quantitative trait.

3.1.4 Gene-gene interaction

Gene-gene interaction is thought to be an important component of the genetic architecture of complex disorders. The involvement of biomolecular interactions in almost every biological process such as gene regulation and metabolism suggests that relationship between DNA sequence variations and clinical endpoints is likely to involve gene-gene interactions.

Epistasis makes the gene-mapping efforts even more daunting as it causes the alteration of the effect of one locus by effects at another locus. In such scenario the power to detect the first locus may get reduced and, in addition, elucidation of the joint effects of the two loci may get hindered by their interaction. If more than two loci are involved, which is the case in most of the complex disorders, the situation is further complicated by the possibility of complex multiway interactions among some or all of the contributing loci.

The following is a summary of the difficulties in the statistical treatment of gene-gene interaction effects[33]:

***Sample size and power*:** Complex traits are influenced by many genetic and non-genetic factors; to detect the compelling and robust evidence of interaction between them a very large sample size is required.

***Modeling interactions*:** Genes can interact in a variety of ways; making assumptions about the specific way in which the interaction manifests itself can be problematic from an analytical point of view, since a large number of parameters might need to be estimated.

***Multiple comparisons*:** If there is no *a priori* knowledge about anticipation, then testing for it would require examination of all possible combinations of the variants used for the study. This number can be as high as tens of thousands for two-locus combinations and may be countless for three or more locus combinations in case of genome-wide scans. Things would further complicate if different assumptions are made for each combination of loci. Multiple testing in such cases raises doubts about false-positive findings.

***Biological significance*:** The detection of statistical interaction does not always imply true gene-gene interaction at the biological level. This is particularly the case in genome-wide scans because the use of marker locus genotypes to draw inferences about putative trait loci could be marred by a number of biological factors which might lead to erroneous inferences about interaction.

3.1.5 Gene-environment interaction

The same genotype at a given trait locus may have different effects on the phenotype under different environmental conditions such as the lifestyle, food habits, surrounding environment etc – a phenomenon known as gene-environment interaction. For example, increased intake of fat and calories and reduced physical activity can lead to obesity, diabetes, asthma, and cardiovascular diseases. However, genetic studies generally ignore the environmental effects and their possible interaction with genes. For example, even though the ε4 allele of *APOE* (*apolipoprotein E*) gene is strongly associated with the common forms of Alzheimer's disease, individuals eating fish at least once a week have been shown to have a 60% lower risk for the disease compared with individuals who never or rarely eat fish[62]. However, how the interplay between fish consumption and *APOE* variation affects the risk of this disease is not clear. Similar effect of environment on the risk of a complex disorder has been demonstrated in the case of schizophrenia, wherein a higher risk of developing this disorder has been shown in children born in winter or spring. Prenatal exposure to viral infections, more prevalent during these seasons, is considered as a possible trigger for the causation of this

effect.[15] Thus, exposure to infectious agents is gaining importance as a likely interacting environmental factor in influencing the risk for complex disorders such as diabetes, heart diseases and obesity.[14]

3.2 STUDYING COMPLEX TRAITS – APPROACHES AND LIMITATIONS

There are myriad methods and approaches available in today's fast-growing field of genetic epidemiology. These can be generally classified as either linkage based or linkage disequilibrium (LD) based approaches. The following are examples of such methods and strategies for studying complex traits.

3.2.1 Linkage-based approaches

Linkage analysis examines the co-segregation of a marker locus with a disease gene locus (i.e. a disease phenotype), with no *a priori* knowledge about the physical position of the disease gene (Figure 2). If the correlation between genotype and phenotype is strong then, when the segregation is evaluated statistically, the probability of finding the true disease gene location is high. However, in cases where some genes have only a minor to moderate effect on the phenotype, as in diseases with multiple disease-causing genes, the probability of finding truly linked loci decreases.

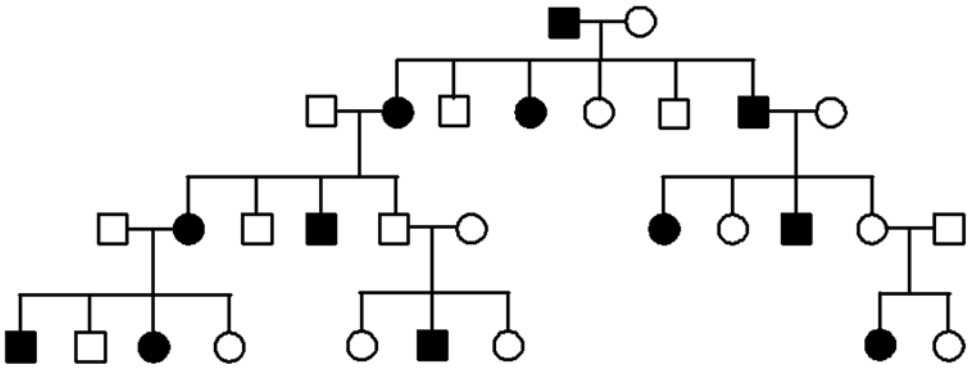

Figure 2. Linkage analysis. Linkage analysis involves studying the co-segregation of genetic markers with the disease in large multi-generational families with multiply affected individuals.

Linkage analysis can be divided into two main strategies. I) parametric (or model-based) linkage analysis and II) non-parametric (or model-free) linkage analysis.

3.2.1.1 Parametric linkage analysis

Parametric linkage analysis uses multi-case families and extended pedigrees. This method involves calculating a logarithm of odds (LOD) score. The LOD score is computed by first calculating the maximum likelihood estimate - the ratio of a marker being linked or not linked to a disease locus under a specific assumption. To conclude whether apparent linkage is "real," the concept of "genome-wide significance" has been developed - the probability threshold that declares linkage after testing many DNA markers used in a genome scan. Lander and Kruglyak[51] suggested 3 levels of genome-wide significance: suggestive linkage, significant linkage and confirmed linkage, though it is suggested that confirmed linkage only occurs when the results are replicated in an independent study sample.

Multipoint linkage analysis is often performed to increase the power of a linkage study. In such type of linkage analysis, several markers known to be located in proximity of each other are analyzed together. If there is true linkage, all markers adjacent to the disease locus are expected to be linked.

Although parametric linkage analysis has the highest power for Mendelian disorders and efficient genome scans can be performed efficiently using this method, it is of limited value for mapping genes for heterogeneous complex disorders. The main drawback of this method is the need to specify the mode of transmission of the disease allele (recessive or dominant), the disease allele frequency in the population and the penetrance value, which are not known in case of most of the complex disorders. However, to get around this, when the data are analyzed several times under different models, the highest LOD score is close to the true mode of

inheritance. Moreover, in the case of many complex disorders, particularly for those with late age at onset (for example Alzheimer's disease), multiply affected large multigenerational families are difficult to collect.

3.2.1.2 Non-parametric linkage analysis

Non-parametric methods avoid problems of selecting a very specific model while still using some of the power of linkage analysis. In these methods affected sib pairs or other pairs of affected relatives are studied and theoretically no assumptions about the disease model are needed to be made. Methods applying this approach are the affected sib-pair method (ASP) using sib-pairs or nuclear families (Figure 3), and the affected pedigree member method (APM) using extended families.

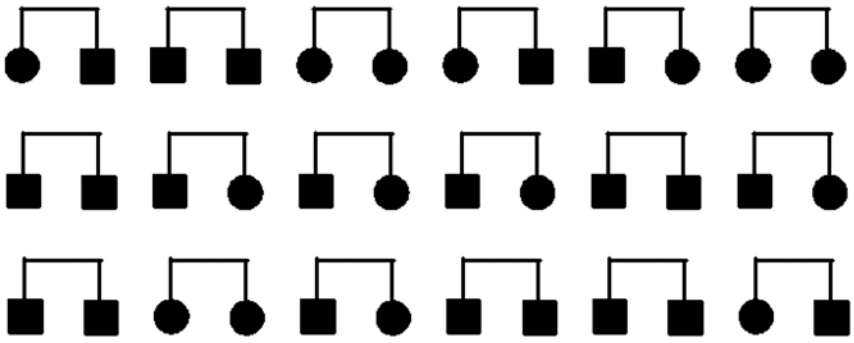

Figure 3. Affected sib-pair (ASP) analysis. ASP analysis involves testing whether affected sib-pairs inherit a region identical-by-decent (IBD) more often than expected under random Mendelian segregation.

The underlying principal behind the non-parametric linkage analysis methods can be explained as follows. Two chromosomal regions are said to be identical-by-descent (IBD) if they

descend from the same ancestral chromosomes. Regions can also be identical-by-state (IBS) if they share the same alleles, but the origin of the region is unknown and therefore the sharing could simply be due to chance. If there is a susceptibility gene located somewhere in the genome and shared by affected individuals IBD, markers physically close to this region will be transmitted along with the disease allele. Furthermore, if a region is shared among affected individuals from the same family more frequently than is expected by random segregation, it may harbor the disease gene. Although, knowing the parental genotypes is not absolutely essential, parental genotypes make such type of analysis more powerful by increasing the certainty about the IBD.

The non-parametric linkage analysis is a method of choice as a first approach for identifying linkage in complex disorders, for example Alzheimer's disease[75] and alcoholism.[73] However, since the microsatellite markers used for genome-wide analysis are widely spaced, the linked chromosomal region may contain hundreds of genes and so to narrow down the list of such positional candidate genes, additional analyses are required.

3.2.2 Linkage disequilibrium (LD) based approaches

LD mapping can be used at the genome-wide level, as a complementary strategy for identifying disease genes within a defined candidate region identified through linkage analysis as well as a candidate gene approach. This approach is based on the fact that alleles at neighboring loci tend to segregate together. If two loci are inherited together more often than would be expected by independent segregation, the two loci are said to be in linkage disequilibrium. It has been suggested that association studies are more powerful than linkage analysis when searching for susceptibility genes for complex disorders. LD mapping can be a useful tool when

performing fine mapping of regions identified by traditional linkage analysis or in the identification of genes of minor effect in complex disorders.

This approach is applied at the population level to perform case-control association analysis, and at the nuclear family level to perform transmission disequilibrium test (TDT) or haplotype relative risk (HRR) analysis. Both case-control and family-based association tests can be performed using single markers or haplotypes.

3.2.2.1 Case-control association studies

The case-control association studies consider specific markers and the allele frequencies between patients and controls are compared (Figure 4). A positive association finding for a specific marker could occur if it was very close to the true disease locus.

Though this approach is considered to be statistically very powerful, case-control association study can generate false positives as a result of population stratification. This problem can be addressed by the use of family-based association methods (discussed below).

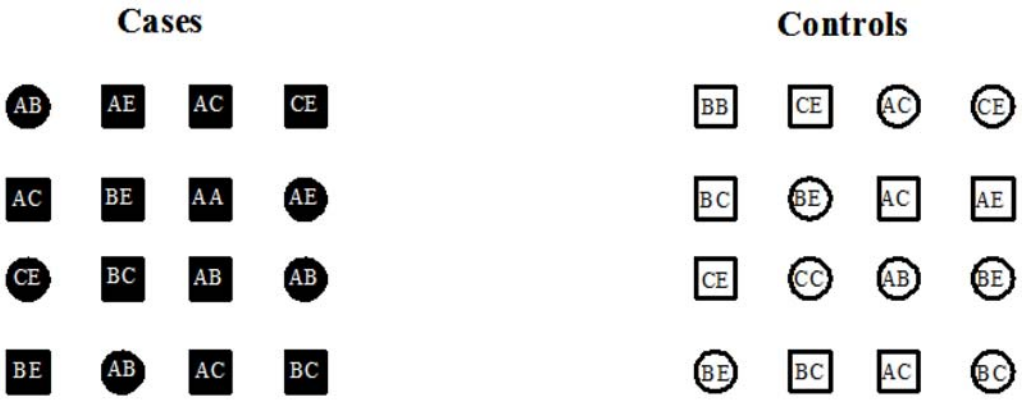

Figure 4. Case-control association analysis (from Burmeister, 1999). Marker allele frequencies are statistically compared between cohorts of unrelated patients and ethnically matched unrelated controls.

3.2.2.2 Family-based association tests

Transmission disequilibrium test (TDT) and haplotype relative risk (HRR) test analyze for preferential transmission of specific alleles (Figure 5). If an association between a genetic marker and a disease locus exists, transmission of the marker locus from parents to an affected offspring would deviate from the expected 0.5 value predicted by Mendelian inheritance. In TDT, only heterozygous parents are considered,[85] while in the HRR test, both homozygous and heterozygous parents are included.[21,89]

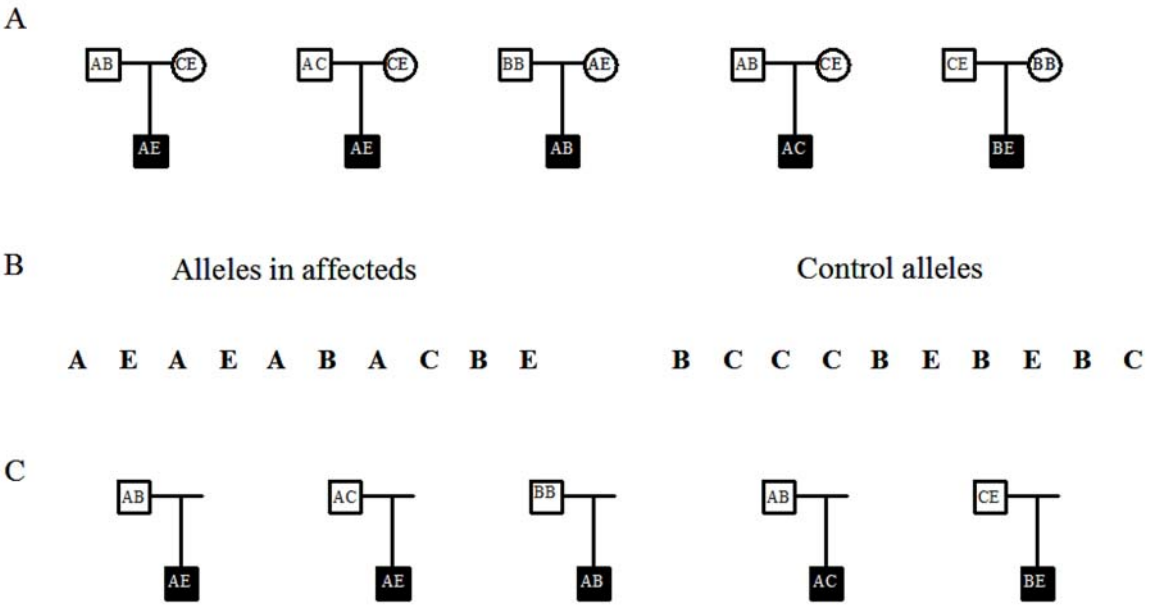

Figure 5. Family-based association analysis (from Burmeister, 1999). A. Family based association analyses start with ascertaining the affected individuals and their parents. **B.** Haplotype relative risk (HRR) method. The allele that is not present in the patient is used as control. **C.** Transmission disequilibrium test (TDT). The preferential transmission of an allele from a heterozygous parent to the affected offspring is tested.

Other transmission test methods include pedigree disequilibrium test (PDT),[56] which incorporates information from all members of a pedigree with genotypic and phenotypic data, and affected family-based controls (AFBAC) test, where both simplex and multiplex families can be ascertained.[91] The AFBAC test uses family data to estimate control marker frequencies assuming random mating in the population. It uses the parental marker alleles not transmitted to an affected child, or never transmitted to an affected sib-pair, as the control population. As a result, association due to ethnic mismatching between patients and controls is avoided, which might cause biased results in traditional case-control studies with unrelated individuals.

3.2.2.3 Haplotype-based approaches

Studying the disease-associated mutations in conjunction with the ancestral haplotypes from which they arose can be statistically powerful methods to apply in association studies of human diseases. Haplotypes created from multiple marker sites can be valuable due to their higher information value relative to single SNPs. The use of haplotype-based approaches has contributed to the identification of genes for both Mendelian disorders[33] and for diseases with a more complex inheritance.[12, 41] Individual haplotypes that contain a mutation will be inherited along with the chromosomal region immediately surrounding that mutation. The size of the ancestral segment shared by the haplotypes is affected by the age of the mutation. Alleles carrying a younger mutation tend to share larger segments than older ones. Studies of haplotype block structure show that only a few haplotypes (approximately 3-5) will constitute the majority (around 90%) of all chromosomes present in a population.[25] Most of these frequently found haplotypes could be distinguished by only a few "key" SNPs, also called the haplotype tag SNPs (htSNP).[12, 42, 67] This suggests that haplotype analysis will reinforce LD mapping by significantly reducing the required number of genotypes, making this a more cost-effective approach.[42]

3.3 INTRODUCTION TO GENETIC ANALYSIS WORKSHOP

The Genetic Analysis Workshops (GAWs) were started in 1982. The main objective of the workshops is to bring together investigators from different parts of the world to put their ideas of research to a single data set. Such an effort was expected to determine the numerical accuracy of the algorithms, to examine the robustness of the methodologies and to compare the range of conclusions from such a data set. Complex traits have been one of the areas of focus of these workshops.

Participants of these workshops are encouraged to interact and cooperate in discussing the various issues related to the data set they worked on, to compare results and interpretations, thereby throwing light on unresolved problems in genetic analysis. These workshops are an excellent place for both beginners and experienced personnel in the research field to gets hands on experience in genetic analysis.

Simulated Data from Genetic Analysis Workshop (GAW) 14

The simulated data for GAW 14 includes an imaginary disease that is thought to be genetic in origin with some environmental factors.

The main characteristics of the genetic model in the simulated dataset that led us to investigate it further are:

1. interaction and
2. heterogeneity.

Four populations were generated (Table 1), each with the same underlying population parameters. Three of the populations had nuclear families. One population had large extended

families. The simulated data set was structured in such a manner that it was possible for participants to work on "fine" mapping and to potentially detect association of the disease due to linkage disequilibrium.

Table 1. Structure of the GAW populations.

Population	Number of pedigrees	Nature of pedigrees
Aipotu (AI)	100	Nuclear families
Danacaa (DA)	100	Nuclear families
Karangar (KA)	100	Nuclear families
New York City (NYC)	50	Extended families

The fine-mapping data were distributed from a specially-designed website that limited the number of SNPs that could be "bought" using an imaginary fund to make it look like a real situation in the management of research funds. The chromosomal regions were packaged into groups of 20 contiguous SNPs ("packets") and participants were allowed to download a maximum of 20 packets. Each packet contained files for all replicates from all the populations studied.

In order to simulate linkage disequilibrium, an area of the genome was used in which there was some LD and two-SNP haplotype probabilities from that region were used to simulate the data.

Collection of Data

The simulated data include data collections from four different (imaginary) groups of investigators from the (simulated) countries of Aipotu (AI), Danacaa (DA), Karangar (KA) and New York City (NYC) (Table 2).

Table 2. Ascertainment scheme used to construct the datasets.

Population	**Affection status based on phenotype**
AI	Affected if either P1, P2 or P3 is present
DA	Affected only if P1 is present
KA	Affected if either P2 or P3 is present
NY	Affected if either P1, P2 or P3 is present

Each of the three groups from the simulated countries of AI, DA and KA collected 100 nuclear families diagnosed with the simulated disease while the NYC group collected 50 extended families with the disease. The summary of the disease model and the locations of the disease-related loci are given in Tables 3 & 4 respectively.

Table 3. Disease model summary.

Phenotype	Major loci involved	Disease allele frequency	Inheritance at loci Epistatic models	Penetrance
P1	D1	0.015	Dominant-Dominant	Penetrance of genotype is 0.6
	D2	0.15		
P2	D2	0.15	D2-D3 Recessive-Dominant D3-D4 Dominant-Recessive	If D6 has allele 1, penetrance is 0.3, otherwise it is 0.6
	D3	0.2		
	D4	0.3		
P3	D1	0.015	D1-D4 Dominant-Recessive	Penetrance of genotype is 1.0
	D4	0.3		
	D2	0.15	D2-D3 Dominant-Recessive	Penetrance of genotype is 0.4
	D3	0.2		

Table 4. Location of disease related loci.

Disease locus	Located between markers
D1	C01R0052 and B01T0561
D2	B03T3067 and C04R0282
D3	B05T4136 and C05R0380
D4	C09R0765 and B09T8337

4.0 METHODS

4.1 OVERVIEW

The simulated dataset from GAW14 was analyzed in order to get a fair idea about how common methods of linkage analysis perform on complex disorders. Since gene interaction and heterogeneity had been incorporated into the genetic model of the simulated disorder in GAW14 and our main objective was to get a better understanding of gene interaction and heterogeneity in complex disorders, this dataset seemed to be very appropriate for the type of questions we wanted to address in complex disorders. As this set of simulated data mimicked a qualitative disorder, we could use methods which are employed in other complex qualitative disorders, such as nonparametric linkage scans, association studies of susceptibility regions, and conditional studies (conditioning on a previously identified susceptibility locus). We tried various methods of conditional analysis like stratification, redefinition, weighted analysis and logistic regression. In this simulated dataset, since it was possible to purchase additional flanking markers for regions of interest following an initial genome scan, we thought that it would be a good exercise to learn how to manage research funds efficiently.

4.2 NONPARAMETRIC LINKAGE ANALYSIS

We began with a standard nonparametric genome scan on the simulated data from replicate 1 with all the microsatellites and SNPs genome-screening markers – 1333 markers in all. We analyzed each of the four populations (Aipotu, Karangar, Danacaa, and NYC) independently. The software Merlin[1] was used to perform the initial genome scans. This yielded linkage signals in four regions on different chromosomes (chromosomes 1, 3, 5, & 9) with suggestive to significant LOD scores with the highest LOD scores on chromosome 3 in all the four populations. In order to perform an in depth analysis on the regions of interest, we purchased sets of markers on chromosomes 1, 3, 5, and 9 flanking the regions with the highest LOD scores from the nonparametric analyses.

4.3 ANALYSES ON THE PURCHASED DATA

4.3.1 Linkage analysis and TDT Analysis (to find the associated alleles)

The additional markers purchased were incorporated into the existing data, and genome scans were repeated on the purchased data set. Minor changes were observed (mostly narrowing the width of the linkage peaks). Follow-up association-based results (transmission disequilibrium test – TDT) were generated with sib_tdt from the ASPEX package[35] for single locus results; while multilocus TDT tests were done with TRANSMIT.[7] The purpose of the follow-up association-based tests was to identify the over-transmitted alleles in the four regions with the highest linkage signals from the initial genome scan analysis. These association-based tests identified the over-transmitted alleles in the four regions with the highest linkage signals. During further analyses, the dataset for the NYC population could not be run properly as the analyses repeatedly crashed due to large family size. So, further analyses and comparisons were restricted to the other three populations.

4.3.2 Interaction Analyses

In order to investigate optimal methods for conditioning linkage analyses, we tried a number of methods. Conditional analysis in genetics can be used to describe the relationship between putative susceptibility genes. Complex disorders involve two or more genes acting together with or without environmental factors. Genome scans may be used to identify single

genes but may not be helpful in detecting the presence of additional genes involved in complex disorders that have an influence over the primary loci. Conditional analysis may be a handy tool in situations where there is a need to differentiate whether the effects are because of heterogeneity or because of interaction. Most methods in conditional analyses aim at stratifying families according to the linkage evidence at a specific locus and analyze them at a second region or genomewide.[11, 47]

Initially it was Gurling et al[29] who reported linkage by using a two-locus model. Smyth et al1[84] were one of the first to use a conditional method involving interaction between two loci in bipolar disorder based on the work done by Gurling et al[29]. Kuida and Beier[48] have shown that (in a murine model) polycystic kidney disease (PKD) progression in the juvenile cystic kidney (jck) mutation can be influenced by interacting modifiers and they localized one of these loci to chromosome 1. Using a chromosome 1 congenic strain, they improved the genetic analysis and mapped the interacting locus to proximal chromosome 4 with a highly significant lod score. Pierik et al[71] studied gene-environment interactions in CD families and found evidence for an interaction between IBD4 and smoking. Linkage was only observed in CD families where at least one of the affected siblings was an active smoker at the time CD was diagnosed. For our analyses on the GAW14 simulated data, we tried some of the existing methods in conditional analyses and some new methods.

4.3.2.1 Stratification

Stratification has been a common procedure used in conditional analysis of complex disorders. The method helps in identifying new regions of linkage thereby supporting epistasis or heterogeneity. Many different ways of stratification have been tried so far in conditional analyses. Stratification by known genotypes, stratification by phenotypes, and stratification by

disease models are some examples. Shaw et al[80] utilized this scheme to find interactive loci with the already determined locus 16q12 (NOD2/CARD15) in a genome-wide search for inflammatory bowel disease susceptibility loci. In this study, the group with the CARD15 variant had suggestive linkage results in 6p and 10p with LOD scores of 3.06 and 2.29 respectively. Hampe et al[31] used stratification by known IBD genotypes. By stratifying on CARD15/NOD2 genotypes, they found evidence for a second IBD locus on chromosome 16p (the IBD8 locus).

We tried stratifying the families based on the overtransmitted alleles in the four above-mentioned regions looking for clues for interaction or heterogeneity. The stratification scheme used was designed to divide the initial data set into two groups – one with families of affected siblings homozygous for the allele of interest and the other, with families of affected siblings heterozygous for the same allele. Since some populations had affected siblings as either all homozygous or all heterozygous for the overtransmitted alleles (all affected individuals in the AI and KA populations were homozygous for allele 1 at B03T3056, and all affected individuals in the DA population were heterozygous for this allele in replicate 1), stratification analyses were not successful in yielding useful results in all populations. The stratification scheme was therefore modified to split each population into two groups – one (named 'present') with families of affected siblings who were homozygous for the allele of interest and the other (named 'absent') with the rest of the families in that population.

The software Merlin[1] was used to carry out nonparametric linkage analysis in each of the stratified pedigree groups. The NYC families are all large extended families as opposed to the other centers, who all collected nuclear families. As Merlin was not able to handle these large pedigrees, Simwalk2 was used to perform nonparametric linkage analysis on the NYC population.

In order to comment on the significance of the results of the split in the LOD scores between the two groups in each population, we decided to do 1000 simulations using the software Allegro. For each simulation, a program was written to randomly split every population into two groups (without replacement), with the number of pedigrees in both the groups similar to the number of pedigrees in the stratified groups. The absolute differences in the LOD scores between these two stratified groups from 1000 simulations were taken into account to calculate the p-value for the LOD score difference obtained during the nonparametric linkage analysis for each stratified group.

4.3.2.2 Weighted analysis

Cox et al, in their study on non-insulin-dependent diabetes (NIDDM), describe an approach to assess the evidence for statistical interactions between unlinked regions that allows multipoint allele-sharing analysis to take the evidence for linkage at one region into account in assessing the evidence for linkage over the rest of the genome.[11] Using this method, they show that the interaction of genes of chromosomes 2 (NIDDM1) and 15 (near CYP19) makes a contribution to susceptibility to type 2 diabetes in Mexican Americans from Starr County, Texas. They used two weighting schemes: in the weight $_{0\text{-}1}$ scheme, families with NPL scores of 0 or negative at NIDDM1 were assigned weight 0 and families with positive NPL scores were assigned weight 1; in the other scheme (weight$_{1\text{-}0}$), they assigned families weight 1 if their NPL score at NIDDM1 was 0 or negative and weight 0 if their NPL score at NIDDM1 was 0 or positive.

A conditional approach has been explored in asthma.[97] Four regions were identified in a genomewide screen (the regions with the highest LOD scores). For each region, analyses were performed by conditioning on that region and searching the rest of the genome for evidence of

additional signals. Based on the evidence for linkage at these four regions, two weighting schemes were used to model a positive and a negative relationship between loci on different chromosomes. In the $weight_{0-1}$ model, a positive relationship (gene-gene interaction) was considered by assigning a weight of 0 to families with a LOD score of 0 or negative and a weight of 1 to families with positive LOD scores. In the $weight_{1-0}$ model (heterogeneity), a negative relationship was modeled. Here, families with negative linkage scores were given a weight of 1 and families with positive linkage scores or a LOD score of 0 were given a weight of 0.

McInnis et al[58] used this scheme to perform conditional analysis in bipolar disorder. Following a primary nonparametric genome-wide scan, they performed conditional analyses based on epistasis or heterogeneity, by weighting the families based on linkage results for the five regions with the highest linkage results. They reported seven potential interactions (four epistasis and three heterogeneity models) in which the NPL scores increased by at least 1.

We started with genome-wide linkage analyses on the dataset using Merlin conditioning on all the four loci with the highest nonparametric LOD scores. For each of the conditioned loci, weights of 1 or 0 were assigned to each family based on the LOD score at the conditional locus. A score of 1 was given to all the pedigrees which had a positive LOD score, and a score of 0 was given to all the pedigrees which had a negative LOD score or a LOD score of 0. The group with the LOD score of 1 was named 'positive' and the group with the LOD score of 0 was called 'negative' in our analyses. And thus we ended up in stratifying the pedigrees according to the presence or absence of linkage to specific regions. Non-parametric linkage analyses were again performed on the stratified pedigree subgroups using Merlin to look for additional positive signals on other regions of the genome.

Two conditional models were examined here – the epistatic model and the heterogeneity model. For the epistatic conditional model, nonparametric linkage analyses were performed on the subgroups with a positive LOD score. Any positive signal on some other region of the genome other than the conditioned locus was considered an interaction since this group already had positive signals for the conditioned locus. For the heterogeneity conditional model, nonparametric linkage analyses were performed on the other subgroups (which were assigned a score of 0). Since these groups did not have any linkage results initially, any positive signal in these groups was considered evidence of heterogeneity.

We also performed permutation tests for each conditional model in all three populations. In each simulation, a program was written to randomly choose a fixed number of pedigrees 1000 times, the number of pedigrees being similar to the number of pedigrees in the subgroups of the weighted analyses. The distribution of LOD scores from the simulations were taken into consideration to calculate the p-value for the LOD score obtained during the nonparametric analysis in each subgroup of the weighted analyses.

4.3.2.3 Redefinition

Redefinition is a concept that has been rarely tried so far by researchers in conditional analyses. Here, instead of splitting the families into groups, the phenotype (affection status) of the individuals is redefined based on the genotypes at regions with significant linkage signals in genomescans. Linkage analyses are repeated after redefining the phenotypes to see how the linkage signals change at the other loci. This redefinition scheme can be used to look for gene-gene interactions as well as gene-environment interactions.

In the redefinition scheme, we redefined the phenotype (affection status) of the individuals based on the genotypes at the four regions of interest mentioned above. The

redefinition procedure was used in two different ways. At first, only individuals who were homozygous for an overtransmitted allele and were previously affected were classified as affected (the 'homoz' group). Next, only individuals who were heterozygous for the overtransmitted allele and were previously affected were classified as affected (the 'heteroz' group). Nonparametric linkage analyses were repeated after redefining the phenotypes to see how the linkage signals changed at the other loci.

4.3.2.4 Logistic Regression

Logistic regression falls under the class of statistical models called generalized linear models (GLM). Logistic regression can be used either to predict group membership or to understand the relationships and strengths among variables.

The model:

While logistic regression makes no assumption about the distribution of the independent or predictor variables, the dependent variable is dichotomous, that is, the dependent variable can take the value 1 with a probability of success *p,* or the value 0 with probability of failure *1-p.* This type of variable is called a binary variable.

For example, if p = the probability of an outcome (e.g., occurrence of a disease) then a possible model could be of the form

$$E(p) = \beta_0+\beta_1X_1+\ldots+\beta_{k-1}X_{k-1} = \underset{1xk}{x} \quad \underset{kx1}{\beta} \qquad (1)$$

where "*E*" stands for expected value, β_0 = *intercept,* β_1 *to* β_{k-1} = *coefficients.*

However, the above model has the property that sometimes the right hand side (RHS) can be <0 or >0. (Recall that probabilities must be between 0 and 1, inclusive.) Since the values on the RHS in (1) can potentially take on values between -∞ and ∞, both sides of the equation have to be transformed so that a very large (or small) value on the RHS of (1) maps into a value between 0 and 1, inclusive. One convenient transformation is that called the *logistic or logit* transformation of *p*. If we first note that the odds measure, p/q = p/1-p, has the property that $0 \leq p/1\text{-}p \leq \infty$, then we can see that if we take the natural log of the odds we have the property that $-\infty < \ln(p/1\text{-}p) < \infty$. Thus, instead of using the model represented in (1), we use

$$logit(p) = ln(p/1\text{-}p) = ln(p/q) = ln(odds) = \beta_0+\beta_1X_1+\ldots+\beta_{k-1}X_{k-1} \qquad (2)$$

North et al applied logistic regression to case-control association studies involving two causative loci, where the two susceptibility loci jointly influenced the risk of developing disease.[64] In their work, simulated case-control samples that were generated assuming different two-locus models were used in the analyses using logistic regression.

We used the *lrmodel* program written by North to implement logistic regression modeling on our dataset. (The notations and formulas used by North et al have been used here to explain how the method works). Our initial nonparametric linkage analysis on the entire dataset yielded four susceptibility regions on different chromosomes – chr1, chr3, chr5, and chr9. When we considered all the possible two-locus interactions among these loci, we came up with six two-locus models for each of the four populations AI, DA, KA and NYC. In each case, a disease model is characterized based on the risk of having the disease conditional on the two-locus genotype. If *i* and *j* are the genotypes at the two loci (where *i, j* = 0, 1 or 2 corresponding to

genotypes 11, 12 and 22 at each locus), then the nine resulting (unphased) multilocus genotypes can be denoted by k (where $k=3i+j$). LRMODEL calculates the multilocus marker genotype frequencies for cases and controls by adding together the genotypings over the total number of cases and controls. If the penetrance for genotype k is termed as f_k and the frequency of genotype k in the population as g_k, then the expected frequency of genotype k in a sample of affected cases is:

$$P^a_k = g_k f_k / K$$

and in a sample of unaffected controls is:

$$p^u_k = g_k(1-f_k)/(1-K)$$

where K is the overall disease prevalence.

A hypothetical dataset consisting of 1000 cases and 1000 controls is got by multiplying these expected frequencies by 1000. LRMODEL fits a sequence of analysis models by means of logistic regression. Each analysis model contains a subset of the full set of genetic effects. If "*r*" is the probability of a subset with the given genotype being a case rather a control, a full model for two loci can be given by:

$$\log(r/1-r) = \mu + a_1x_1 + d_1z_1 + a_2x_2 + d_2z_2 + i_{aa}x_1x_2 + i_{ad}x_1z_2 + i_{da}z_1x_2 + i_{dd}z_1z_2 \quad (3)$$

where
- x and z - dummy variables specific to each locus, defining additive and dominance effects
- μ - mean
- a_1, a_2 - additive effects of the two loci
- d_1, d_2 - dominance effects of the two loci
- i_{aa}, i_{ad}, i_{da}, i_{dd} - interactive effects

The various combinations of these additive, dominance and interactive effects form the basis for the disease models used for the analysis as described by North et al. ADD1, ADD2 and ADD stand for models with additive effects, DOM1, DOM2 and DOM stand for models with dominance effects, and ADDINT, ADDDOM and DOMINT stand for models with interactive effects.

The method provides two measures of adequacy, the Likelihood Ratio Statistic (LRS) and Akaike's Information Criterion (AIC).

$$LRS = 2\ln(L2/L1)$$

$$AIC = -2\ln(L)+b$$ {b is the number of free parameters in the model}

The AIC measure is a substitute to the LRS for model assessment. A model with the lowest AIC value is regarded to have the best fit and most parsimonious.

5.0 RESULTS

5.1 INITIAL LINKAGE ANALYSIS

Nonparametric genome scans on the simulated data from replicate 1 yielded interesting NPL results for markers on chromosomes 1, 3, 5, and 9 with LOD scores greater than 1.5. A particular region on chromosome 3 showed significant linkage with suggestive (>2.0) or significant (>3.5) LOD scores in all populations. As we were considering these populations to be independent studies, we considered similar results in different populations reflective of replication. Based on these criteria, loci on chromosomes 1 and 3 are considered "confirmed", while loci on chromosomes 5 and 9 are only "suggested" (Table 5).

Table 5. Results from genome scan with microsatellite and SNP genome scan markers.

Population (# of individuals)	***LOD (cM)**			
	Chr1	**Chr3**	**Chr5**	**Chr9**
AI (783)	1.72 (22)	3.85 (87.295)		
DA (700)	6.48 (48.5)	2.15 (89.295)		1.97 (53.806)
KA (694)		5.89 (95.295)	5.19 (1.806)	5.18 (2.306)
NY (943)	2.4 (54)	2.73 (94.795)		

*Results are presented as npl LOD scores and locations at which the maximum score occurred. Only LOD scores > 1.5 are presented.

5.2 ANALYSES ON THE PURCHASED DATA

5.2.1 Linkage Analysis and Association Analysis

After purchasing relevant data sets based on the results of the primary analyses, genome scans were repeated on the purchased data set (after incorporating the additional markers into the existing data). The results are provided in Figure 6.

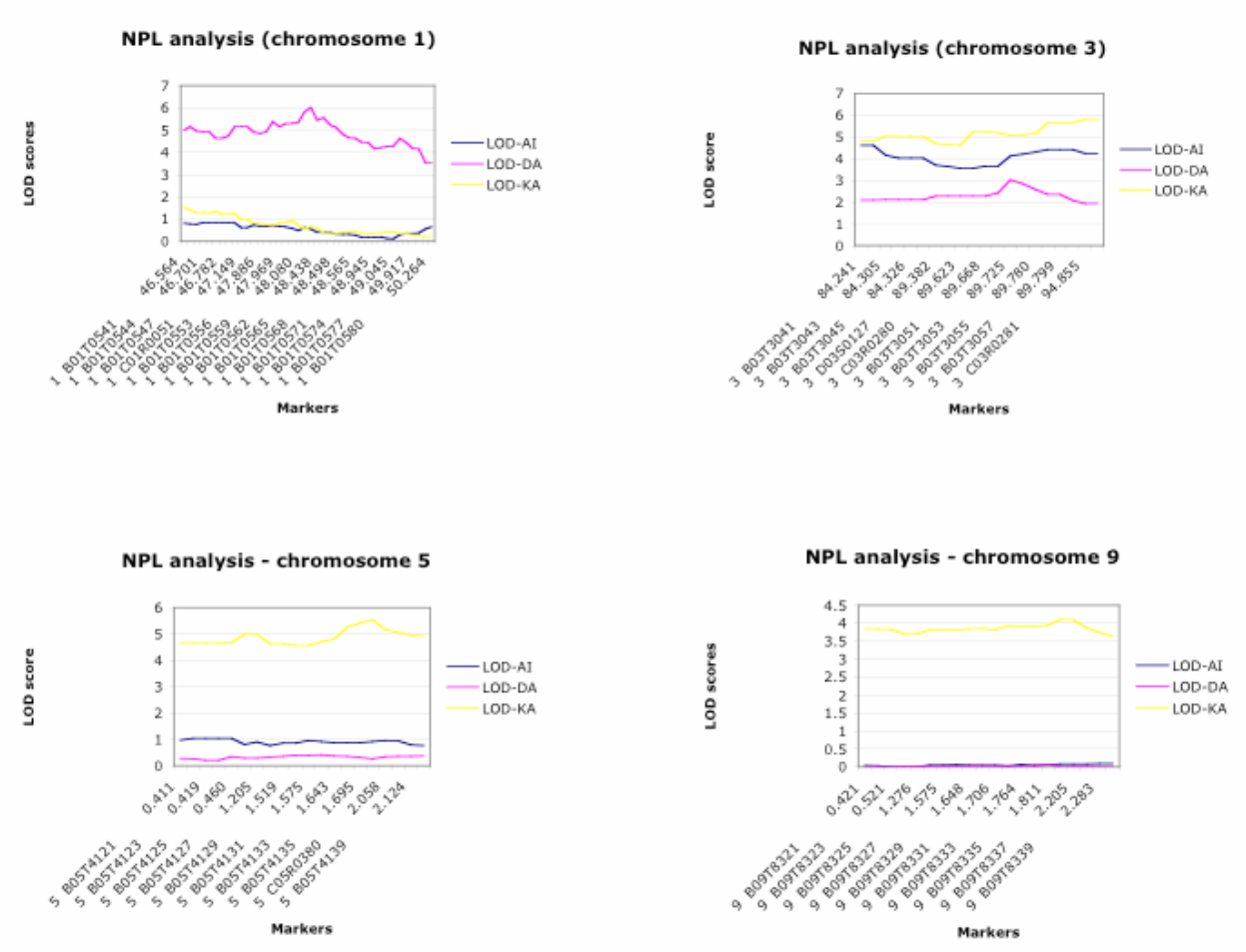

Figure 6. Comparison of LOD scores from npl analyses on purchased data – by chromosome and by population

Transmission disequilibrium tests on the purchased data set using ASPEX (sib_tdt) and TRANSMIT packages yielded significant single locus chi-squared scores for each population. The region on chromosome 3 (snp B03T3056) had the greatest chi-squared scores (ranging between 3.56 and 34.16) for all the populations. The TDT test also showed that in all these populations, for marker B03T3056 on chromosome 3, allele 1 was overtransmitted by a ratio of 2.5:1. Similar results were obtained when all four populations were combined. We also used TRANSMIT to test for multi-locus association between the genetic markers of interest on these chromosomes and the disease, to see if there were particular haplotypes which were associated. In so doing, we identified multi(2)-locus haplotypes which exhibited evidence of linkage to the disease phenotype, but with significance levels no greater than the single-locus tests. The overtransmitted alleles for the four regions with the highest linkage signals are listed in Table 6. Most of the conditioning in the future analyses was based on these overtransmitted alleles.

Table 6. Overtransmitted alleles for regions with highest linkage signals on the genome.

Chromosome	Locus	Allele #
1	B01T0558	1
3	B03T3056	1
5	B05T4140	2
9	B09T8333	2

The above mentioned analyses and tests were carried out on replicate 2, which yielded similar results.

5.2.2 Interaction Analysis

As one of our primary objectives was to comprehend how heterogeneity and interaction operate in complex disorders, further analyses were focused on conditional methods that could help detect heterogeneity and interaction in the complex disorder represented in GAW14 simulated data set. Also efforts were made to compare how these methods perform with respect to one another. As described earlier, four different conditional methods were employed and nonparametric linkage analyses were carried out to see how the linkage signals behaved at the conditioned loci in different populations under different methods. Also, in order to examine the effect of sample size, the conditional analyses were repeated on bigger samples by combining some of the replicates. Replicates 1 and 2 were combined to double the sample size and replicates 1 through 5 were combined to produce an even bigger sample size.

5.2.2.1 Stratification

The pedigrees in each population were split into two groups based on any affected sibling being homozygous for an overtransmitted allele as described earlier. The results of the nonparametric analyses from each of the stratified groups by population are laid out in Tables 7, 8, 9 and 10. These tables also contain results from initial nonparametric linkage analysis on the full sets of pedigrees for a quick comparison.

Table 7. Nonparametric analyses in the stratified groups – Replicate 1.

Based on	Pop		# of aff. Fam.	LOD scores chr 1 46.841-	chr 3 89.725-	chr 5 1.757-	chr 9 1.721-
chr1	ai	present	38	0.02	0.71	-0.05	0.04
		absent	62	0.13	0	0.01	0.04
		all	100	0.69	4.24	0.92	0.08
	da	present	46	4.35	1.72	-0.02	0
		absent	54	1.69	2.56	0.64	0
		all	100	5.56	3.02	0.26	0
	ka	present	35	0.02	1.61	0.4	3.4
		absent	65	0.47	4.18	5.5	1.5
		all	100	0.39	5.8	5.53	4.1
chr3	ai	present	71	0.63	7.64	0.65	-0.07
		absent	29	no result	no result	no result	no result
		all	100	0.69	4.24	0.92	0.08
	ka	present	69	0	9.01	2.53	0.06
		absent	31	no result	no result	no result	no result
		all	100	0.39	5.8	5.53	4.1
chr5	ai	present	69	1.4	2.69	0.24	0.16
		absent	31	0.02	2.33	0.7	0
		all	100	0.69	4.24	0.92	0.08
	da	present	27	3.71	0.56	-0.05	-0.01
		absent	73	4.06	1.97	0.61	0.4
		all	100	5.56	3.02	0.26	0
	ka	present	84	0.22	4.66	5.92	4.92
		absent	16	0.21	0.85	0.22	0.02
		all	100	0.39	5.8	5.53	4.1
chr9	ai	present	26	0.31	0.01	0.1	0.1
		absent	74	0.18	6.58	0.85	0.4
		all	100	0.69	4.24	0.92	0.08
	da	present	33	1.2	2	0.23	-0.72
		absent	67	6.1	0.88	0.09	0.47
		all	100	5.56	3.02	0.26	0
	ka	present	48	1.7	1.35	3.5	4
		absent	52	0	5.5	2.35	1.38
		all	100	0.39	5.8	5.53	4.1

Table 8. Nonparametric analyses in the stratified groups – Replicate 2.

Based on	Pop		# of aff. Fam.	LOD scores chr 1 46.841-	chr 3 89.725-	chr 5 1.757-	chr 9 1.721-
chr1	ai	present	42	-0.03	0.76	0.3	0.21
		absent	58	1.69	0.05	1.68	1.33
		all	100	0.73	0.57	1.85	1.41
	da	present	42	2.52	1.72	0.33	0.03
		absent	58	1.36	1.41	0	0.05
		all	100	3.58	3.01	0.13	0.09
	ka	present	36	-0.03	0.86	0.52	2.63
		absent	64	0	0.54	5.61	2.45
		all	100	-0.01	1.32	5.52	4.92
chr3	ai	present	57	0.23	0.29	1.77	0.17
		absent	43	0.54	0.29	0.3	1.59
		all	100	0.73	0.57	1.85	1.41
	da	present	63	1.7	2.24	0.39	0.11
		absent	37	1.97	0.8	-0.06	0
		all	100	3.58	3.01	0.13	0.09
	ka	present	69	-0.71	1	4.49	1.69
		absent	31	1.51	0.34	1.31	3.89
		all	100	-0.01	1.32	5.52	4.92
chr5	ai	present	47	0.55	0.32	0.42	2.04
		absent	53	0.23	0.26	1.68	0.13
		all	100	0.73	0.57	1.85	1.41
	da	present	46	0.32	1.05	-1.28	0.22
		absent	54	3.97	2	2.63	0
		all	100	3.58	3.01	0.13	0.09
	ka	present	38	-0.16	1.36	1.08	2.64
		absent	62	0.03	0.35	4.6	2.33
		all	100	-0.01	1.32	5.52	4.92
chr9	ai	present	30	0.56	0.73	0.88	0
		absent	70	0.31	0.1	0.99	1.84
		all	100	0.73	0.57	1.85	1.41
	da	present	40	2.99	1.26	0.11	-1.54
		absent	60	1.08	1.75	0.04	1.91
		all	100	3.58	3.01	0.13	0.09
	ka	present	30	-0.54	2.16	3.6	-0.06
		absent	70	0.12	0.14	2.34	7.1
		all	100	-0.01	1.32	5.52	4.92

Table 9. Nonparametric analyses in the stratified groups – Combined replicates 1 and 2.

				LOD scores			
Based on	**Pop.**		**# of aff. Fam.**	**chr 1 46.841-48.451 cM**	**chr 3 89.725-94.881 cM**	**chr 5 1.757-2.181 cM**	**chr 9 1.721-2.154 cM**
chr1	ai	present	81	-0.01	1.69	0.07	0.19
		absent	119	2.32	2.5	3.33	1
		all	200	1.3	4.19	2.49	1.12
	da	present	89	5.76	2.34	0.17	0.09
		absent	111	3.41	3.03	0.3	0.02
		all	200	8.82	5.35	0.47	0.1
	ka	present	71	-0.13	2.27	0.95	5.77
		absent	129	0.97	3.9	10.29	3.61
		all	200	0.37	6.16	10.42	8.76
chr3	ai	present	128	0.46	2.26	2.56	0.12
		absent	72	1.03	2.01	0.28	1.47
		all	200	1.3	4.19	2.49	1.12
	da	present	132	5.49	3.51	0.6	0
		absent	68	3.37	1.84	0	0.28
		all	200	8.82	5.35	0.47	0.1
	ka	present	138	-0.13	4.76	7.37	2.27
		absent	62	2.85	1.52	3.05	8.98
		all	200	0.37	6.16	10.42	8.76
chr5	ai	present	91	0.85	2.07	0.06	0.89
		absent	109	0.49	2.13	3.95	0.35
		all	200	1.3	4.19	2.49	1.12
	da	present	95	3.76	2.93	-1.03	0.02
		absent	105	5.1	2.47	3.9	0.09
		all	200	8.82	5.35	0.47	0.1
	ka	present	85	0.04	4.53	2.7	4.68
		absent	115	0.4	2.18	7.91	4.13
		all	200	0.37	6.16	10.42	8.76
chr9	ai	present	73	2.19	5.57	2.01	-0.11
		absent	127	0.13	0.57	0.77	2.26
		all	200	1.3	4.19	2.49	1.12
	da	present	90	7.38	1.44	0.13	-1.28
		absent	110	2.34	4.12	0.36	2.14
		all	200	8.82	5.35	0.47	0.1
	ka	present	57	0	5.06	3.79	-0.55
		absent	143	0.53	2.25	6.64	14.84
		all	200	0.37	6.16	10.42	8.76

Table 10. Nonparametric analyses in the stratified groups – Combined replicates 1 through 5.

				LOD scores			
				chr 1	**chr 3**	**chr 5**	**chr 9**
Based on	**Pop**		**# of aff. Fam.**	**46.841-48.451 cM**	**89.725-94.881 cM**	**1.757-2.181 cM**	**1.721-2.154 cM**
chr1	ai	present	193	1.41	5.51	0.95	0.8
		absent	307	6.36	9.08	6.03	3.96
		all	500	7.47	14.59	6.42	4.48
	da	present	209	13.02	6.4	0.27	0.28
		absent	291	20.44	9.88	0.88	0.9
		all	500	33.45	16.28	1.1	1.13
	ka	present	185	-0.24	3.63	2.81	9.71
		absent	315	1.82	6.25	22.97	6.66
		all	500	0.66	9.88	23.21	15.82
chr3	ai	present	319	2.85	10.7	6.57	0.48
		absent	181	5.39	4.01	0.73	5.92
		all	500	7.47	14.59	6.42	4.48
	da	present	324	22.15	11.46	0.74	0.31
		absent	176	11.3	5.01	0.35	1.11
		all	500	33.45	16.28	1.1	1.13
	ka	present	325	-0.01	8.6	20.23	2.14
		absent	175	2.36	1.81	4.2	21.63
		all	500	0.66	9.88	23.21	15.82
chr5	ai	present	229	6.68	7.76	0.06	1.92
		absent	271	1.75	6.85	10.18	2.56
		all	500	7.47	14.59	6.42	4.48
	da	present	226	13.6	8.57	-1.93	0.28
		absent	274	19.97	7.83	6.54	0.91
		all	500	33.45	16.28	1.1	1.13
	ka	present	209	0.13	6.3	6.4	9.43
		absent	291	0.57	4.13	17.27	6.81
		all	500	0.66	9.88	23.21	15.82
chr9	ai	present	184	4.82	10.85	3.72	-0.12
		absent	316	3.08	5.43	2.9	8.07
		all	500	7.47	14.59	6.42	4.48
	da	present	224	20.2	7.9	0.04	-1.53
		absent	276	13.97	8.41	1.5	5.76
		all	500	33.45	16.28	1.1	1.13
	ka	present	162	0.21	8.05	14.11	-0.65
		absent	338	0.45	3.43	10.46	26.87
		all	500	0.66	9.88	23.21	15.82

In replicate 1, while stratifying based on allele 1 for the locus B03T3056 on chromosome 3, we found that there were no informative affected pairs for the disease in the Aipotu and Karangar populations in the 'absent' groups. As a result, there was no output for these groups in these two populations. For the same replicate, in the Danacaa population, all the affected individuals were heterozygous for allele 1 (locus B03T3056). So, it was not possible to have any families in the 'present' group to carry out the stratification. It was noted that the LOD scores in most of the stratified groups were unevenly distributed in all the populations.

Tables 11, 12, 13 and 14 contain the results of the simulations done on the stratified groups. LOD scores greater than 5.0 with significant p-values were seen in a few stratified groups.

Table 11. Simulations on stratified groups – Replicate 1.

Based on	**chr 1**		**chr 3**		**chr 5**		**chr 9**	
	diff	**p-value**	**diff**	**p-value**	**diff**	**p-value**	**diff**	**p-value**
chr 1								
ai	0.11	0.964	0.71	0.732	0.06	0.98	0	1
da	2.66	0.128	0.84	0.633	0.66	0.736	0	1
ka	0.45	0.807	2.57	0.093	5.1	**0.002**	1.9	0.25
chr 5								
ai	1.38	0.704	0.36	0.917	0.54	0.879	0.16	0.969
da	0.35	0.941	0.41	0.926	0.66	0.877	0.41	0.926
ka	0.01	0.984	3.81	0.013	5.7	0.001	4.9	0.005
chr 9								
ai	0.13	0.956	6.57	**0**	0.75	0.634	0.3	0.872
da	4.9	0.024	1.12	0.672	0.14	0.954	1.19	0.65
ka	1.7	0.307	4.15	0.016	1.15	0.497	2.62	0.103

P-values indicative of heterogeneity and interaction are printed in bold and underlined respectively. NR- No results; ND- Not done

Table 12. Simulations on stratified groups – Replicate 2.

Based on		chr 1		chr 3		chr 5		chr 9	
		diff	**p-value**	**diff**	**p-value**	**diff**	**p-value**	**diff**	**p-value**
chr 1									
	ai	1.72	0.069	0.71	0.355	1.65	0.074	1.12	0.178
	da	1.16	0.563	0.31	0.9	0.33	0.893	0.02	0.995
	ka	0.03	0.981	0.32	0.813	4.91	**0**	0.18	0.909
chr 3									
	ai	0.31	0.712	0	1	1.47	0.105	1.42	0.112
	da	0.27	0.817	1.16	0.327	0.45	0.704	0.11	0.924
	ka	2.22	0.031	0.66	0.358	3.18	0.003	2.2	0.033
chr 5									
	ai	0.32	0.664	0.06	0.946	1.26	0.132	1.91	0.036
	da	3.65	**0.007**	0.95	0.56	3.91	**0.003**	0.22	0.911
	ka	0.19	0.876	1.01	0.381	3.52	**0.004**	0.31	0.798
chr 9									
	ai	0.25	0.782	0.63	0.473	0.11	0.902	1.84	0.033
	da	1.91	0.182	0.49	0.833	0.07	0.981	3.45	**0.009**
	ka	0.66	0.834	2.02	0.37	1.26	0.644	7.16	**0**

P-values indicative of heterogeneity and interaction are printed in bold and underlined respectively. NR- No results; ND- Not done

Table 13. Simulations on stratified groups – Combined replicates 1 and 2.

Based on		chr 1		chr 3		chr 5		chr 9	
		diff	**p-value**	**diff**	**p-value**	**diff**	**p-value**	**diff**	**p-value**
chr 1									
	ai	2.33	0.235	0.81	0.726	3.26	0.083	0.81	0.726
	da	2.35	0.187	0.69	0.737	0.13	0.944	0.07	0.973
	ka	1.1	0.595	1.77	0.372	9.36	**0**	2.16	0.257
chr 3									
	ai	0.57	0.908	0.25	0.957	1.28	0.763	1.35	0.749
	da	2.12	0.503	1.67	0.637	0.6	0.884	0.28	0.945
	ka	2.98	0.361	3.24	0.302	4.32	0.079	6.71	**0**
chr 5									
	ai	0.36	0.914	0.06	0.986	3.89	0.036	0.54	0.86
	da	1.34	0.476	0.46	0.824	4.93	0.016	0.07	0.968
	ka	0.36	0.897	2.35	0.267	5.21	0.015	0.55	0.836
chr 9									
	ai	2.06	0.29	5	0.002	1.28	0.554	2.37	0.205
	da	5.04	0.007	2.68	0.136	0.23	0.901	3.42	0.062
	ka	0.53	0.768	2.81	0.107	2.85	0.097	14.29	**0**

P-values indicative of heterogeneity and interaction are printed in bold and underlined respectively. NR- No results; ND- Not done

Table 14. Simulations on stratified groups – Combined replicates 1 through 5.

Based on	chr 1		chr 3		chr 5		chr 9	
	diff	**p-value**	**diff**	**p-value**	**diff**	**p-value**	**diff**	**p-value**
chr 1								
ai	4.95	0.315	3.57	0.523	5.08	0.293	3.88	0.479
da	7.42	0.806	3.48	0.924	0.61	0.983	0.62	0.983
ka	2.06	0.327	2.62	0.212	20.16	**0**	2.95	0.161
chr 3								
ai	2.54	0.362	6.69	0.028	5.84	0.054	5.44	0.076
da	10.85	0.073	6.45	0.295	0.39	0.963	0.8	0.925
ka	2.37	0.472	6.79	0.002	16.03	0	19.51	**0**
chr 5								
ai	4.93	0.165	0.91	0.847	10.12	**0.001**	0.64	0.878
da	6.37	0.749	0.74	0.975	8.47	0.634	0.63	0.979
ka	0.44	0.82	2.17	0.298	10.87	**0**	2.62	0.214
chr 9								
ai	1.74	0.838	5.42	0.276	0.82	0.924	8.19	0.056
da	6.23	0.766	0.51	0.979	1.46	0.951	7.29	0.71
ka	0.24	0.93	4.62	0.046	3.65	0.123	27.52	**0**

P-values indicative of heterogeneity and interaction are printed in bold and underlined respectively. NR- No results; ND- Not done

For replicate 1, simulations were not possible for the locus on chromosome 3 (B03T3056) as either there were inadequate informative affected pairs for the disease in the AI and KA populations, or there were no homozygous affected individuals for the same locus in the DA population. While comparing the results of the simulations with the expected number of interactions in the GAW dataset, we observed that in all the replicates the number of false positives was greater than the number of true positives.

5.2.2.2 Weighted Analysis

After splitting the families into those which had positive LOD scores and those which had negative LOD scores or a LOD score of 0 at the conditioned loci from an initial genomescan, follow-up analyses were carried out on these subgroups as described below. A total of 24

additional nonparametric linkage analyses were done on these subgroups for each set of replicates (single or combined). The results of these linkage analyses are presented in tables 15, 16, 17 and 18.

To determine the significance of the results of the conditional analyses, we performed 1000 simulations for each of the subgroups. The results of the simulations are presented in tables 15, 16, 17 and 18. With the help of the simulation analyses, we found out that this method also did not work effectively on the given simulated dataset in detecting interaction or heterogeneity (very low percentage of true positives).

Table 15. Nonparametric analyses and simulations in the weighted groups – Replicate 1.

				LOD scores							
				chr 1		chr 3		chr 5		chr 9	
Based on	**Pop**	**Sub-group**	**# of aff. Fam.**	**46.841-48.451 cM**	**Sim. p-value**	**89.725-94.881 cM**	**Sim. p-value**	**1.757-2.181 cM**	**Sim. p-value**	**1.721-2.154 cM**	**Sim. p-value**
chr1	ai	positive	52	10.21	**0**	3.58	0.181	0.29	0.9	0.01	1
		negative	47	-2.18	1	1.14	0.748	0.58	0.972	0.05	1
		all	100	0.69		4.24		0.92		0.08	
	da	positive	64	13.76	**0**	1.52	0.871	0.21	1	-0.13	1
		negative	36	-1.91	1	0.83	1	0.06	1	0.28	1
		all	100	5.56		3.02		0.26		0	
	ka	positive	59	10.78	**0**	4.81	0.14	5.08	<u>0.009</u>	2.36	0.462
		negative	41	-2.06	1	1.25	0.892	0.94	0.97	1.74	0.648
		all	100	0.39		5.8		5.53		4.1	
chr3	ai	positive	59	1.48	0.84	15.53	**0**	0.73	0.971	0	1
		negative	40	0	1	-0.95	1	0.25	1	0.29	1
		all	100	0.69		4.24		0.92		0.08	
	da	positive	54	2.36	0.743	12.6	**0**	0.28	1	0	1
		negative	46	2.42	0.849	-1.86	1	0.04	1	0.01	1
		all	100	5.56		3.02		0.26		0	
	ka	positive	59	1.32	0.821	14.5	**0**	1.95	0.587	1.83	0.628
		negative	41	0.16	1	-1.28	1	3.78	0.064	2.41	0.347
		all	100	0.39		5.8		5.53		4.1	
chr5	ai	positive	53	0.19	1	1.97	0.773	10.89	**0**	0	1
		negative	47	0.8	0.917	2.5	0.217	-1.23	1	0.18	1
		all	100	0.69		4.24		0.92		0.08	
	da	positive	47	1.44	0.991	0.65	1	10.09	**0**	0.26	1
		negative	53	3.41	0.413	1.83	0.887	-2.64	1	-0.22	1
		all	100	5.56		3.02		0.26		0	
	ka	positive	61	1.02	0.883	1.97	0.58	13.74	**0**	1.83	0.63
		negative	39	0.26	1	4.67	0.035	-1.21	1	2.38	0.373
		all	100	0.39		5.8		5.53		4.1	
chr9	ai	positive	42	0.43	0.991	1.82	0.712	0.1	1	8.49	**0**
		negative	58	0.41	1	2.59	0.283	0.96	0.935	-1.34	1
		all	100	0.69		4.24		0.92		0.08	
	da	positive	42	1.25	0.961	0.91	0.99	0.6	1	8.34	**0**
		negative	58	3.49	0.427	1.44	0.999	0	1	-2.78	1
		all	100	5.56		3.02		0.26		0	
	ka	positive	64	0.45	0.988	3.6	0.126	2.09	0.53	15.25	**0**
		negative	36	0.89	0.99	2.05	0.549	3.52	0.103	-2.46	1
		all	100	0.39		5.8		5.53		4.1	

P-values indicative of heterogeneity and interaction are printed in bold and underlined respectively. NR- No results; ND- Not done

Table 16. Nonparametric analyses and simulations in the weighted groups – Replicate 2.

Based on	Pop	Sub-group	# of aff. Fam.	LOD scores chr 1 46.841-48.451 cM	sim. p-value	chr 3 89.725-94.881 cM	sim. p-value	chr 5 1.757-2.181 cM	sim. p-value	chr 9 1.721-2.154 cM	sim. p-value
chr1	ai	positive	47	10.06	0	0.96	0.537	3.38	0.006	0.77	0.706
		negative	53	-2.22	1	0.02	1	0.06	1	0.65	0.722
		all	100	0.73		0.57		1.85		1.41	
	da	positive	57	13	0	1.29	0.936	0.2	1	0.01	1
		negative	43	-2.07	1	1.81	0.837	0	1	0.09	1
		all	100	3.58		3.01		0.13		0.09	
	ka	positive	47	8.86	0	0.07	0.997	2.3	0.079	2.3	0.079
		negative	53	-4.27	1	1.85	0.037	3.24	**0.004**	2.68	**0.01**
		all	100	-0.01		1.32		5.52		4.92	
chr3	ai	positive	45	2.5	0.042	10.8	0	1.75	0.126	0.85	0.659
		negative	55	-0.03	1	-2.56	1	0.42	0.888	0.58	0.783
		all	100	0.73		0.57		1.85		1.41	
	da	positive	60	2.18	0.425	14.27	0	-0.02	1	0	1
		negative	40	1.4	0.896	-3.7	1	0.59	0.998	0.22	1
		all	100	3.58		3.01		0.13		0.09	
	ka	positive	51	-0.02	1	11.9	0	4.54	0	1.04	0.507
		negative	49	0	1	-3.07	1	1.48	0.065	4.29	**0.001**
		all	100	-0.01		1.32		5.52		4.92	
chr5	ai	positive	57	1.94	0.069	0.32	0.977	12.25	0	0.37	0.971
		negative	43	-0.17	1	0.25	0.939	-2.76	1	1.3	0.139
		all	100	0.73		0.57		1.85		1.41	
	da	positive	42	2.76	0.085	0.84	0.94	9.3	0	0.14	1
		negative	58	1.19	0.998	2.21	0.812	-3.82	1	0	1
		all	100	3.58		3.01		0.13		0.09	
	ka	positive	63	-0.06	1	0.67	0.634	13.61	0	1.19	0.308
		negative	37	0.02	1	0.79	0.338	-1.61	1	4.89	**0**
		all	100	-0.01		1.32		5.52		4.92	
chr9	ai	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	100	0.73	ND	0.57	ND	1.85	ND	1.41	ND
		all	100	0.73		0.57		1.85		1.41	
	da	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	100	3.58	ND	3.01	ND	0.13	ND	0.09	ND
		all	100	3.58		3.01		0.13		0.09	
	ka	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	100	-0.01	ND	1.32	ND	5.52	ND	4.92	ND
		all	100	-0.01		1.32		5.52		4.92	

P-values indicative of heterogeneity and interaction are printed in bold and underlined respectively. NR- No results; ND- Not done

Table 17. Nonparametric analyses in the weighted groups – Combined replicates 1 and 2.

				LOD scores							
				chr 1		chr 3		chr 5		chr 9	
Based on	**Pop**	**Sub-group**	**# of aff. Fam.**	**46.841-48.451 cM**	**sim. p-value**	**89.725-94.881 cM**	**sim. p-value**	**1.757-2.181 cM**	**sim. p-value**	**1.721-2.154 cM**	**sim. p-value**
chr1	ai	positive	97	19.87	0	3.91	0.205	2.4	0.744	0.53	1
		negative	103	-4.48	1	0.95	0.917	0.58	0.995	0.58	0.995
		all	200	1.3		4.19		2.49		1.12	
	da	positive	121	28.03	0	2.88	0.999	0.24	1	-0.2	1
		negative	79	-4.01	1	2.5	0.972	0.24	1	0.9	1
		all	200	8.82		5.35		0.47		0.1	
	ka	positive	97	19.25	0	2	0.705	4.77	0.038	3.41	0.211
		negative	103	-5.4	1	4.35	**0.008**	5.66	**0.002**	5.49	**0.002**
		all	200	0.37		6.16		10.42		8.76	
chr3	ai	positive	104	3.09	0.56	26.33	0	2.02	0.874	0.4	1
		negative	96	-0.01	1	-2.28	1	0.65	0.974	0.75	0.955
		all	200	1.3		4.19		2.49		1.12	
	da	positive	114	4.69	0.679	26.87	0	0.04	1	0	1
		negative	86	4.18	0.43	-3.76	1	0.66	1	0.21	1
		all	200	8.82		5.35		0.47		0.1	
	ka	positive	110	0.3	0.999	26.4	0	5.78	0.007	2.59	0.587
		negative	90	0.09	0.999	-3.6	1	4.64	**0.004**	6.8	**0**
		all	200	0.37		6.16		10.42		8.76	
chr5	ai	positive	113	1.3	0.977	1.99	0.908	23.76	0	0.15	1
		negative	87	0.16	1	2.23	0.128	-2.43	1	1.46	0.443
		all	200	1.3		4.19		2.49		1.12	
	da	positive	92	4.28	0.586	1.66	0.997	19.83	0	0.34	1
		negative	108	4.56	0.539	3.8	0.866	-5.26	1	-0.02	1
		all	200	8.82		5.35		0.47		0.1	
	ka	positive	124	0.37	1	3.17	0.388	26.7	0	3.5	0.275
		negative	76	0.04	1	3.25	0.031	-2.45	1	5.6	**0.004**
		all	200	0.37		6.16		10.42		8.76	
chr9	ai	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	200	1.3	ND	4.19	ND	2.49	ND	1.12	ND
		all	200	1.3		4.19		2.49		1.12	
	da	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	200	8.82	ND	5.35	ND	0.47	ND	0.1	ND
		all	200	8.82		5.35		0.47		0.1	
	ka	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	200	0.37	ND	6.16	ND	10.42	ND	8.76	ND
		all	200	0.37		6.16		10.42		8.76	

P-values indicative of heterogeneity and interaction are printed in bold and underlined respectively. NR- No results; ND- Not done

Table 18. Nonparametric analyses in the weighted groups – Combined replicates 1 through 5.

				LOD scores							
Based on	**Pop**	**Sub-group**	**# of aff. Fam.**	**chr 1 46.841-48.451 cM**	**sim. p-value**	**chr 3 89.725-94.881 cM**	**sim. p-value**	**chr 5 1.757-2.181 cM**	**sim. p-value**	**chr 9 1.721-2.154 cM**	**sim. p-value**
chr1	ai	positive	258	57.6	0	7.68	0.104	4.28	0.915	2.05	1
		negative	242	-9.96	1	6.93	0.579	2.27	1	2.46	1
		all	500	7.47		14.59		6.42		4.48	
	da	positive	332	75.93	0	10.42	0.999	0.37	1	0.04	1
		negative	168	-7.37	1	5.86	1	0.85	1	2.29	1
		all	500	33.45		16.28		1.1		1.13	
	ka	positive	226	47.83	0	3.13	0.478	7.04	0.005	5.99	0.025
		negative	274	-13.34	1	7.11	**0**	16.64	**0**	9.99	**0**
		all	500	0.66		9.88		23.21		15.82	
chr3	ai	positive	272	5.45	0.661	66.93	0	5.91	0.513	1.06	1
		negative	228	2.25	1	-4.94	1	1.25	1	3.89	0.988
		all	500	7.47		14.59		6.42		4.48	
	da	positive	283	18.86	0.091	68.07	0	0.53	1	0.18	1
		negative	217	14.64	0.861	-8.44	1	0.57	1	1.2	1
		all	500	33.45		16.28		1.1		1.13	
	ka	positive	252	0.46	0.999	60.43	0	13.12	0	5.29	0.074
		negative	248	0.22	1	-10.07	1	10.17	**0**	11.03	**0**
		all	500	0.66		9.88		23.21		15.82	
chr5	ai	positive	279	4.48	0.939	8.8	0.064	59.18	0	0.64	1
		negative	221	2.99	0.997	5.84	0.764	-5.97	1	5.2	0.858
		all	500	7.47		14.59		6.42		4.48	
	da	positive	231	13.63	0.226	6.24	1	50.57	0	1.79	1
		negative	269	19.86	0.697	10.26	1	-13.43	1	0.03	1
		all	500	33.45		16.28		1.1		1.13	
	ka	positive	303	0.45	1	6.63	0.006	67.38	0	5.72	0.028
		negative	197	0.21	1	3.26	0.116	-6.61	1	11.19	**0**
		all	500	0.66		9.88		23.21		15.82	
chr9	ai	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	500	7.47	ND	14.59	ND	6.42	ND	4.48	ND
		all	500	7.47		14.59		6.42		4.48	
	da	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	500	33.45	ND	16.28	ND	1.1	ND	1.13	ND
		all	500	33.45		16.28		1.1		1.13	
	ka	positive	0	NR	ND	NR	ND	NR	ND	NR	ND
		negative	500	0.66	ND	9.88	ND	23.21	ND	15.82	ND
		all	500	0.66		9.88		23.21		15.82	

P-values indicative of heterogeneity and interaction are printed in bold and underlined respectively. NR- No results; ND- Not done

5.2.2.3 Redefinition

The results of the analyses based on redefining the phenotypes of the original dataset are summarized in tables 19, 20, 21 and 22. A quick glance at the nature of the distribution of the alleles at the conditioned loci from this table showed that except for the marker B03T3056 on chromosome 3, every other marker that we conditioned on had their alleles distributed in a heterozygous fashion in the majority of the cases in all the populations. Only marker B03T3056 on chromosome 3 had some tendency of being distributed in the homozygous fashion in most of the cases in all the populations.

Table 19. Nonparametric analyses in the redefined groups – Replicate 1.

Based on	Pop	Sub-group	# of affecteds	LOD scores chr 1 46.841-48.451 cM	chr 3 89.725-94.881 cM	chr 5 1.757-2.181 cM	chr 9 1.721-2.154 cM
chr1	ai	homoz	78	0.45	0.72	0.31	-0.03
		heteroz	145	2.2	1.92	3.43	-0.13
		all	281	0.69	4.24	0.92	0.08
	da	homoz	97	**7.46**	0	-0.4	-0.29
		heteroz	118	3.04	2.19	0.48	0
		all	258	5.56	3.02	0.26	0
	ka	homoz	56	1.22	0.48	0.21	3.52
		heteroz	135	3	4.23	4.21	1.05
		all	242	0.39	5.8	5.53	4.1
chr3	ai	homoz	160	0.1	**7.64**	0.65	-0.07
		heteroz	111	1.73	3.41	0	0.21
		all	281	0.69	4.24	0.92	0.08
	da	homoz	0	no result	no result	no result	no result
		heteroz	107	2.42	1.69	0.15	0.08
		all	258	5.56	3.02	0.26	0
	ka	homoz	136	0	**9.01**	2.53	0.06
		heteroz	87	1.12	1.3	2.01	3.77
		all	242	0.39	5.8	5.53	4.1
chr5	ai	homoz	67	0.37	0.89	4.3	-0.14
		heteroz	130	0.01	0.63	3.01	1.48
		all	281	0.69	4.24	0.92	0.08
	da	homoz	50	0.67	0.33	2.58	-0.05
		heteroz	116	0.77	0.29	1.09	0.08
		all	258	5.56	3.02	0.26	0
	ka	homoz	50	1.87	1.84	2.39	0.03
		heteroz	102	-0.06	0.43	4.35	1.73
		all	242	0.39	5.8	5.53	4.1
chr9	ai	homoz	55	0.28	0.06	0.01	1.51
		heteroz	142	-0.07	2.79	0	1.36
		all	281	0.69	4.24	0.92	0.08
	da	homoz	59	0.02	0.35	1.84	0.31
		heteroz	108	0.38	1.61	0.37	0.51
		all	258	5.56	3.02	0.26	0
	ka	homoz	94	0.26	0.13	1.98	**7.31**
		heteroz	104	0	3.22	1.6	3.18
		all	242	0.39	5.8	5.53	4.1

LOD scores indicative of heterogeneity are printed in bold.

Table 20. Nonparametric analyses in the redefined groups – Replicate 2.

Based on	Pop	Sub-group	# of affecteds	LOD scores chr 1 46.841-8.451 cM	chr 3 89.725-94.881 cM	chr 5 1.757-2.181 cM	chr 9 1.721-2.154 cM
chr1	ai	homoz	77	1.01	0.78	1.44	0.37
		heteroz	119	2.77	0.07	0.76	0.78
		all	256	0.73	0.57	1.85	1.41
	da	homoz	81	**5.1**	0.9	0.19	-0.11
		heteroz	116	1.3	0.28	0.35	0.15
		all	253	3.58	3.01	0.13	0.09
	ka	homoz	68	1.52	0.15	0.52	1.51
		heteroz	132	0.73	0.12	2.49	0.94
		all	258	-0.01	1.32	5.52	4.92
chr3	ai	homoz	125	-0.05	1.99	0.78	0.11
		heteroz	93	0.85	2.2	0.65	0.09
		all	256	0.73	0.57	1.85	1.41
	da	homoz	127	0.1	**5.37**	0.19	0.55
		heteroz	119	2.63	2.22	-0.13	-0.02
		all	253	3.58	3.01	0.13	0.09
	ka	homoz	145	-0.33	**4.31**	5.78	0.5
		heteroz	79	0.44	0.53	1	1.87
		all	258	-0.01	1.32	5.52	4.92
chr5	ai	homoz	47	0.02	0.01	0.62	0.01
		heteroz	117	0.19	0.15	3.22	0.2
		all	256	0.73	0.57	1.85	1.41
	da	homoz	53	0.61	1.85	2.43	0.43
		heteroz	130	2.63	0.74	2.13	-0.04
		all	253	3.58	3.01	0.13	0.09
	ka	homoz	36	-0.01	0.07	2.23	-0.01
		heteroz	140	0.1	0.56	4.02	1.85
		all	258	-0.01	1.32	5.52	4.92
chr9	ai	homoz	71	0.15	-0.03	0.27	**5.11**
		heteroz	128	1.1	0.1	0.7	1.54
		all	256	0.73	0.57	1.85	1.41
	da	homoz	63	0.57	0.58	0.07	0.73
		heteroz	115	0.21	0.38	0	1.72
		all	253	3.58	3.01	0.13	0.09
	ka	homoz	83	0.37	-0.06	1.75	**6.07**
		heteroz	116	-0.01	0.49	0.58	2.69
		all	258	-0.01	1.32	5.52	4.92

LOD scores indicative of heterogeneity and interaction are printed in bold and underlined respectively.

Table 21. Nonparametric analyses in the redefined groups – Combined replicates 1 and 2.

				LOD scores			
Based on	**Pop**	**Sub-group**	**# of affecteds**	**chr 1 46.841-48.451 cM**	**chr 3 89.725-94.881 cM**	**chr 5 1.757-2.181 cM**	**chr 9 1.721-2.154 cM**
chr1	ai	homoz	155	1.68	1.45	1.2	0.13
		heteroz	264	4.27	1.49	2.37	0.29
		all	537	1.3	4.19	2.49	1.12
	da	homoz	179	**12.25**	0.54	-0.01	-0.37
		heteroz	233	3.16	1.88	0.91	0.08
		all	511	8.82	5.35	0.47	0.1
	ka	homoz	124	2.5	0.51	0.66	4.27
		heteroz	267	3.11	2.62	5.66	2
		all	500	0.37	6.16	10.42	8.76
chr3	ai	homoz	285	0.1	8.81	1	0.02
		heteroz	204	1.91	5.33	0.41	0.19
		all	537	1.3	4.19	2.49	1.12
	da	homoz	271	0.88	**12.08**	0.86	0.13
		heteroz	228	4.83	3.79	0	0.07
		all	511	8.82	5.35	0.47	0.1
	ka	homoz	281	-0.02	**12.55**	6.98	0.41
		heteroz	166	1.13	1.64	2.33	5.29
		all	500	0.37	6.16	10.42	8.76
chr5	ai	homoz	114	0.31	0.5	4.53	-0.12
		heteroz	247	0.41	0.73	6.22	1.24
		all	537	1.3	4.19	2.49	1.12
	da	homoz	103	1.04	1.91	5.01	0.14
		heteroz	246	2.3	0.99	4.02	0.01
		all	511	8.82	5.35	0.47	0.1
	ka	homoz	85	1.05	1.02	4.62	0.35
		heteroz	242	0.03	0.99	8.22	3.54
		all	500	0.37	6.16	10.42	8.76
chr9	ai	homoz	126	0.39	0	0.14	5.27
		heteroz	270	0.19	1.26	0.32	2.7
		all	537	1.3	4.19	2.49	1.12
	da	homoz	122	0.37	0.86	0.6	1.02
		heteroz	223	0.72	1.47	0.17	2.31
		all	511	8.82	5.35	0.47	0.1
	ka	homoz	175	0.87	0.01	3.2	**13.35**
		heteroz	220	0	2.77	2	5.73
		all	500	0.37	6.16	10.42	8.76

LOD scores indicative of heterogeneity and interaction are printed in bold and underlined respectively.

Table 22. Nonparametric analyses in the redefined groups – Combined replicates 1 through 5.

				LOD scores (chromosome/cM)			
Based of	**Pop**	**Sub-group**	**# of affecteds**	**chr 1 46.841-48.451 cM**	**chr 3 89.725-94.881 cM**	**chr 5 1.757-2.181 cM**	**chr 9 1.721-2.154 cM**
chr1	ai	homoz	365	**11.16**	2.77	1.73	0.6
		heteroz	654	11.74	5.25	2.67	1.2
		all	1332	7.47	14.59	6.42	4.48
	da	homoz	423	**23.9**	2.28	-0.09	-0.27
		heteroz	592	17.71	5.54	0.49	0.5
		all	1258	33.45	16.28	1.1	1.13
	ka	homoz	326	5.07	1.41	0.85	7.05
		heteroz	658	5.76	4.07	11.08	3.17
		all	1272	0.66	9.88	23.21	15.82
chr3	ai	homoz	711	1.02	**26.89**	4.68	0.06
		heteroz	504	4.93	8.95	1.88	1.45
		all	1332	7.47	14.59	6.42	4.48
	da	homoz	641	5.9	**29.37**	2.52	0.56
		heteroz	574	13.5	9.84	0.01	0.14
		all	1258	33.45	16.28	1.1	1.13
	ka	homoz	659	-0.02	**27.17**	16.19	0.23
		heteroz	454	1.03	4.61	1.85	11.4
		all	1272	0.66	9.88	23.21	15.82
chr5	ai	homoz	270	0.4	1.6	**10.53**	-0.21
		heteroz	618	1.34	3.49	15.83	3.87
		all	1332	7.47	14.59	6.42	4.48
	da	homoz	239	2.05	2.92	**7.33**	0.34
		heteroz	631	13.02	3.77	9.25	0.92
		all	1258	33.45	16.28	1.1	1.13
	ka	homoz	224	0.18	3.31	**7.92**	1.84
		heteroz	628	0.04	0.89	20.13	3.86
		all	1272	0.66	9.88	23.21	15.82
chr9	ai	homoz	319	1.24	-0.02	1.91	**13.41**
		heteroz	654	1.78	9.21	0.57	7.53
		all	1332	7.47	14.59	6.42	4.48
	da	homoz	263	1.6	1.99	0.4	4.81
		heteroz	596	9.06	4.18	1.16	**8.22**
		all	1258	33.45	16.28	1.1	1.13
	ka	homoz	379	0.67	0.05	1.85	**26.22**
		heteroz	599	0.15	3.56	4.01	16.87
		all	1272	0.66	9.88	23.21	15.82

LOD scores indicative of heterogeneity and interaction are printed in bold and underlined respectively.

5.2.2.4 Logistic Regression

As one of the methods of conditioning, a logistic regression approach was implemented as described previously. For each of the four populations, logistic regression analysis was performed on six two-locus interaction models. The summary of the results is displayed in table 23.

The table gives the AIC values for the dominant disease model and also the AIC difference. The AIC difference is calculated by taking the difference between the least AIC score for that interaction and the AIC score for a particular model. The model with the least AIC value is considered to have the best fit.

As most of the interaction models incorporated into the simulated data set included a dominance effect at least at one of the two interacting loci, we concentrated on the 'DOM' model in interpreting the output from the logistic regression analysis using the program LRMODEL. The other models in the output either talk about effects at single loci or about models not present in our data set.

Table 23. Logistic Regression analyses using the program LRModel.

	1&3		1&5		1&9		3&5		3&9		5&9	
Pop	**AIC**	**AIC Diff**	**AIC**	**AIC Diff**	**AIC**	**AIC Diff**	**AIC**	**AIC Diff**	**AIC**	**AIC Diff**	**AIC**	**AIC Diff**
Replicate 1 (DOM model)												
AI	210.28	4.18	227.48	5.05	225.33	3.22	211.86	5.76	209.01	3.31	226.99	4.88
DA	165.70	1.19	175.04	3.96	176.62	5.54	167.06	2.54	168.14	3.62	177.28	6.23
KA	138.09	0	148.13	3.83	146.74	2.44	145.46	5.62	144.99	5.15	155.78	7.01
NY	51.01	1.75	85.20	5.93	83.77	4.49	54.43	5.18	50.92	2.50	85.42	6.15
Replicate 2 (DOM model)												
AI	177.55	2.29	193.44	5.95	194.26	7.34	177.04	1.78	178.07	2.81	193.35	5.86
DA	183.27	2.95	194.54	7.05	189.24	3.86	184.14	3.83	176.33	0	188.94	3.55
KA	168.88	3.44	180.98	6.46	181.31	6.8	168.88	3.45	168.72	3.29	180.42	5.91
NY	85.6	3.69	88.72	5.72	86.52	3.53	86.63	4.72	84.4	2.49	86.95	3.96
Combined replicates 1&2 (DOM model)												
AI	380.26	0.95	413.59	3.16	412.68	5.65	382.57	3.26	379.87	2.4	413.53	3.78
DA	341.22	1.07	361.03	4.24	359.64	2.84	342.76	2.61	338.43	0	359.32	2.28
KA	301.74	0	324.78	3.61	323.62	2.45	306.54	6.66	306.38	3.44	328.75	6.43
Combined replicates 1 thru 5 (DOM model)												
AI	974.06	3.71	1031.76	4.64	1035.6	5.77	972.2	2.89	973.76	4.32	1031.3	4.2
DA	844.17	0	876.95	3.96	876.93	3.94	847.86	3.04	846.44	1.62	879.12	4.44
KA	867.4	3.04	899.89	5.41	896.8	3.24	867.01	2.64	862.66	0	895.45	1.89
NY	358.55	4.38	414.52	7.55	410.0	5.54	358.57	4.41	349.75	1.85	410.41	5.95
NY	95.84	5.18	158.92	6.22	156.59	5.81	95.46	4.8	89.77	3.21	155.82	5.05

For each dataset the analysis models which are within an AIC difference of 2 compared with the best model (AIC difference =0) are highlighted.

6.0 DISCUSSION

The stratification of pedigrees based on the homozygosity of alleles at the four loci B01T0558, B03T3056, B05T4140 and B09T8333 reflected the unequal distribution of these genotypes among different populations. The results of the analyses on the stratified pedigree groups indicated the possibility of false positive results in many populations, which led us to carry out simulation tests. As suggested, following simulation tests, the stratification method detected a very low percentage of true positives at a type I error (α) level of 0.01. When we look at how the method performed for each of the four populations separately, the results are not significant. For the DA population, the method performed overly conservative because it did not detect any interaction at all (with neither true positives nor false positives). For both the AI and the KA populations, false positive rates were greater than true positive rates.

In the weighted analysis, there was a more even distribution of the genotypes between the 'positive' and 'negative' groups as opposed to the uneven representation in the stratified analyses. Nonparametric linkage analyses on the subgroups yielded very significant signals in all the populations at the four loci on chromosomes 1, 3, 5 and 9. But careful comparison of those interactions with the GAW 'answers' showed that most of those were false positive signals.

Both the redefinition method and the logistic regression method have a population-based approach instead of a family-based approach (as in stratification method or weighted analysis). In the redefinition method, individuals were considered as cases only if they had a particular

combination of alleles at the loci they were being conditioned on. In the logistic regression method, irrespective of the genotypes at the two interested loci, cases were those individuals who were originally affected in the populations. Logistic regression analysis was performed for each of the six two-locus interaction models possible in each of the four populations. Also logistic regression was the only analysis which was able to handle the huge pedigree sizes of the New York City population. The results from these analyses showed a very low power in the detection of the expected interactions. Comparing with each other, the logistic regression method seemed to be more meaningful than the redefinition method.

During the process of this study, we learned that every single method used had its advantages and disadvantages. Some analyses were easy to perform with a very short running time but the results were hard to interpret. The results of logistic regression are a good example of this problem (Table 23). While others took an eternity to run, the results were very clear (refer to tables 11 through 18 for the results of simulations on AI, DA and KA populations). Also, we understood that all the weaknesses of a method should be addressed and taken care of in order to obtain better results. This was true when we discovered that increasing the sample size alone did not increase the power in our analyses (Tables 7 through 22), while the other inadequacies of the methods were not addressed.

Researchers have always struggled to find the appropriate cut-off for LOD scores in interpreting their results even with the availability of guidelines on the significance of LOD scores by Lander and Kruglyak[51]. In this light, we find the weighted analysis method more convincing since it groups all the families with any positive LOD score into one set for the conditional analysis, thereby minimizing the chances of missing any relevant information from the results. A quick glance at Table 24 will tell us that the weighted analysis method did better in

detecting more true positives in three of the four methods used. Also, the subgroups have a more even distribution of pedigrees (or individuals) in the weighted analysis as compared to other methods, which we attribute to the low cut-off LOD score of the weighted analysis method.

By and large, as evident from the results from the methods used, it is possible to say that these methods were successful in detecting heterogeneity more than interactions. In some cases, the signals for heterogeneity may be interpreted as suggestive of allelic heterogeneity rather than locus heterogeneity. For example, while conditioning on chromosome 9 (B09T8333) in Replicate 2 in the stratification method (Table 12), we get significant p-values in DA and KA populations at the same locus, which can be suggestive of allelic heterogeneity.

7.0 CONCLUSION

The main idea of this project was to study various methods of analyses in a dataset simulating a complex disease and comment on how these methods perform in detecting interaction or heterogeneity. A genome scan was successfully carried out on the simulated dataset of GAW14. Transmission Disequilibrium Test was used to identify loci with overtransmitted alleles. Further analyses were performed conditioning on these overtransmitted alleles using stratification method, redefinition method, weighted analysis method and logistic regression. The methods that we used did not prove to be very effective in detecting interaction or heterogeneity. Each of the methods we tried provided more false positive than true positive results (Table 24).

We believe that a number of reasons contributed to this fact as mentioned earlier in the background section. In our simulated dataset, incomplete penetrance, naïve assumption of the marker for conditioning, very narrow range of flanking markers and genetic heterogeneity (possibly allelic heterogeneity) are considered important factors in not achieving a high degree of performance.

Table 24. Comparison of results of conditional analyses with the GAW 'answers'.

Stratification			Weighted Analysis			Redefinition			Logistic Regression		
Exp.	Observed		Exp.	Observed		Exp.	Observed		Exp.	Observed	
	TP (%)	FP (%)		TP (%)	FP (%)		TP (%)	FP (%)		TP (%)	FP (%)
Replicate 1											
12	1 (8.3)	3 (25)	16	0	1 (25)	15	0	1 (6.7)	12	2 (16.7)	1 (18.3)
Replicate 2											
16	1 (6.25)	6 (37.5)	12	3 (25)	3 (25)	16	1 (6.25)	0	12	1 (8.3)	1 (8.3)
Combined replicates 1 and 2											
16	0	5 (31.3)	12	3 (25)	3 (25)	16	1 (6.25)	0	12	2 (16.7)	2 (16.7)
Combined replicates 1 through 5											
16	1 (6.25)	6 (37.5)	12	4 (33)	3 (25)	16	2 (12.5)	0	12	2 (16.7)	3 (25)

We also calculated the prevalence of interactions in the GAW14 simulated dataset from the disease allele frequencies (Table 25). This gave us a pretty clear picture of the low prevalence of interactions in this simulated dataset, which explained the inability of the methods to detect interactions. To improve the effectiveness of these methods we suggest that future investigators address these specific issues. Considering haplotype studies instead of targeting single markers, and increasing the range of the flanking markers around regions of high LOD scores could be some of the solutions to the problems encountered so far in this simulated dataset.

Table 25. Prevalence of interactions.

Population	Interaction			Model	Prevalence of interaction
AI	**Chromosomes**	**1**	**3**	D/D	1/2 X 1/2 = 0.0075 1/1 X 1/2 = 0.0000573 1/2 X 1/1 = 0.0006648 1/1 X 1/1 = 0.000005
	Disease allele	1	1		
	Allele frequency	0.015	0.015		
	Chromosomes	**1**	**9**	D/R	1/2 X 2/2 = 0.0026595 1/1 X 2/2 = 0.0000742
	Disease allele	1	2		
	Allele frequency	0.015	0.3		
	Chromosomes	**3**	**5**	D/R or R/D	1/1 X 1/2 = 0.0072 1/1 X 2/2 = 0.0009 or 1/2 X 2/2 = 0.0102 1/1 X 2/2 = 0.0009
	Disease allele	1	2		
	Allele frequency	0.15	0.2		
	Chromosomes	**5**	**9**	D/R	1/2 X 2/2 = 0.0288 2/2 X 2/2 = 0.0036
	Disease allele	2	2		
	Allele frequency	0.2	0.3		
DA	**Chromosomes**	**1**	**3**	D/D	1/2 X 1/2 = 0.0075 1/1 X 1/2 = 0.0000573 1/2 X 1/1 = 0.0006648 1/1 X 1/1 = 0.000005
	Disease allele	1	1		
	Allele frequency	0.015	0.015		
KA	**Chromosomes**	**1**	**9**	D/R	1/2 X 2/2 = 0.0026595 1/1 X 2/2 = 0.0000742
	Disease allele	1	2		
	Allele frequency	0.015	0.3		
	Chromosomes	**3**	**5**	D/R or R/D	1/1 X 1/2 = 0.0072 1/1 X 2/2 = 0.0009 or 1/2 X 2/2 = 0.0102 1/1 X 2/2 = 0.0009
	Disease allele	1	2		
	Allele frequency	0.15	0.2		
	Chromosomes	**5**	**9**	D/R	1/2 X 2/2 = 0.0288 2/2 X 2/2 = 0.0036
	Disease allele	2	2		
	Allele frequency	0.2	0.3		

8.0 FUTURE STUDIES

The main objective of our study was to focus on the comparison of various methods of linkage analyses and conditional analyses in a simulated dataset of a complex disease. Even though interaction and heterogeneity were incorporated as main components into the simulated dataset of GAW14, our methods were not effective in detecting the same efficiently. By using more efficient strategies to carefully choose the regions on the genome for conditional analyses, future studies on this dataset could provide better results using the same methods used in this study.

APPENDIX A

SHELL SCRIPT FOR NON-PARAMETRIC LINKAGE ANALYSIS

```
#!/bin/tcsh -f
# C-shell file name: script.sh
#-----------------------------------------------
#       Input file names:
#        Locus file:            datain.dat
#        Pedigree files:        ai.dat da.dat ka.dat ny.dat
#        Map file:              map.dat
#-----------------------------------------------
mkdir Strat Redef Weight Logist
cd Strat
# STRATIFICATION ANALYSES
# Starting with AI population
# Stratifying based on chr1 locus B01T0558 - allele 1 (columns 41 & 42)
mkdir AI DA KA NY
mkdir AI_results DA_results KA_results NY_results
cd AI
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
```

```
mkdir Present Absent
cd Present
        cp ../../../../ai.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '$3 > 0' ai.dat > tmp1.dat
        awk '$6 > 1' tmp1.dat > tmp2.dat
        awk '$41 < 2' tmp2.dat > tmp3.dat
        awk '$42 < 2' tmp3.dat > tmp4.dat
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ai.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ai.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

        pull_pedigrees.pl ai.dat pedlist1.dat pedin.dat

# running mega2
```

```
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_pre_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_pre_09.lod

    cd ../Absent
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .

# compare ai.dat with pedin.dat in the directory-present, then retain those
# lines that are not in pedin.dat and write it to a file called pedin.dat
    cp ../Present/pedin.dat pedin1.dat
    cat ./pedin1.dat ./ai.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
```

```
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_abs_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_abs_09.lod
```

```
        awk '/B09T8333/' merlin_out.09 > ai_abs_09.lod

cd ../..
# Move into the next : Chr3/
cd Chr3

# Repeating all the steps done for Chr1

mkdir Present Absent
cd Present
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '$3 > 0' ai.dat > tmp1.dat
    awk '$6 > 1' tmp1.dat > tmp2.dat
    awk '$117 < 2' tmp2.dat > tmp3.dat
    awk '$118 < 2' tmp3.dat > tmp4.dat
    awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
    rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ai.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:
```

```
# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ai.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

      pull_pedigrees.pl ai.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

      ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
```

```
# the file merlin_out.01 and write it to the file ai_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_pre_09.lod

    cd ../Absent
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Present/pedin.dat pedin1.dat
    cat ./pedin1.dat ./ai.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
```

```
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_abs_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_abs_03.lod

# print the lines that contain the pattern B05T4140 in
```

```
# the file merlin_out.05 and write it to the file ai_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_abs_09.lod

cd ../..

# Move into the next: Chr5

cd Chr5

# Repeat steps

mkdir Present Absent
cd Present
    cp ../../../../ai.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
    awk '$3 > 0' ai.dat > tmp1.dat
    awk '$6 > 1' tmp1.dat > tmp2.dat
    awk '$165 < 2' tmp2.dat > tmp3.dat
    awk '$166 < 2' tmp3.dat > tmp4.dat
    awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
    rm tmp*
```

```
# if the number in the pedlist.dat matches the number in the first field
# of the file ai.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ai.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

      pull_pedigrees.pl ai.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
```

```
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_pre_09.lod

    cd ../Absent
```

```
        cp ../../../../ai.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        cp ../Present/pedin.dat pedin1.dat
        cat ./pedin1.dat ./ai.dat | sort | uniq -u > pedin.dat
             awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

        ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_abs_01.lod
```

```
        awk '/B01T0558/' merlin_out.01 > ai_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_abs_09.lod

cd ../..

# Move into the next : Chr9

cd Chr9

# Repeat steps

mkdir Present Absent
cd Present
    cp ../../../../ai.dat .
```

```
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '$3 > 0' ai.dat > tmp1.dat
        awk '$6 > 1' tmp1.dat > tmp2.dat
        awk '$191 < 2' tmp2.dat > tmp3.dat
        awk '$192 < 2' tmp3.dat > tmp4.dat
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ai.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ai.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

        pull_pedigrees.pl ai.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
```

```
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_pre_05.lod
```

```
            awk '/B05T4140/' merlin_out.05 > ai_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_pre_09.lod

            awk '/B09T8333/' merlin_out.09 > ai_pre_09.lod

      cd ../Absent
      cp ../../../../ai.dat .
      cp ../../../../datain.dat .
      cp ../../../../map.dat .
      cp ../Present/pedin.dat pedin1.dat
      cat ./pedin1.dat ./ai.dat | sort | uniq -u > pedin.dat
           awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input
```

```
./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_abs_01.lod

awk '/B01T0558/' merlin_out.01 > ai_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_abs_03.lod

awk '/B03T3056/' merlin_out.03 > ai_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_abs_05.lod

awk '/B05T4140/' merlin_out.05 > ai_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_abs_09.lod

awk '/B09T8333/' merlin_out.09 > ai_abs_09.lod

# Move up by three levels

cd ../../..
```

```
# Move into the AI_results directory and copy the results only

cd AI_results

mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1

cp ../../AI/Chr1/Present/pedlist1.dat .

cp ../../AI/Chr1/Present/ai_pre* .

cp ../../AI/Chr1/Absent/pedlist2.dat .

cp ../../AI/Chr1/Absent/ai_abs* .


cd ../Chr3

cp ../../AI/Chr3/Present/pedlist1.dat .

cp ../../AI/Chr3/Present/ai_pre* .

cp ../../AI/Chr3/Absent/pedlist2.dat .

cp ../../AI/Chr3/Absent/ai_abs* .


cd ../Chr5

cp ../../AI/Chr5/Present/pedlist1.dat .

cp ../../AI/Chr5/Present/ai_pre* .

cp ../../AI/Chr5/Absent/pedlist2.dat .

cp ../../AI/Chr5/Absent/ai_abs* .


cd ../Chr9

cp ../../AI/Chr9/Present/pedlist1.dat .

cp ../../AI/Chr9/Present/ai_pre* .

cp ../../AI/Chr9/Absent/pedlist2.dat .

cp ../../AI/Chr9/Absent/ai_abs* .
```

```
cd ../../

# Move into the next population DA and repeat the analyses.
cd DA
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
mkdir Present Absent
cd Present
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '$3 > 0' da.dat > tmp1.dat
    awk '$6 > 1' tmp1.dat > tmp2.dat
    awk '$41 < 2' tmp2.dat > tmp3.dat
    awk '$42 < 2' tmp3.dat > tmp4.dat
    awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
    rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file da.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
```

```
#   awk '$1 == '$ped" da.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

      pull_pedigrees.pl da.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

      ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_pre_01.lod
```

```
        awk '/B01T0558/' merlin_out.01 > da_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > da_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > da_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > da_pre_09.lod

    cd ../Absent
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .

# compare ai.dat with pedin.dat in the directory-present, then retain
# those
# lines that are not in pedin.dat and write it to a file called pedin.dat
    cp ../Present/pedin.dat pedin1.dat
    cat ./pedin1.dat ./da.dat | sort | uniq -u > pedin.dat
```

```
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_abs_01.lod

        awk '/B01T0558/' merlin_out.01 > da_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_abs_03.lod
```

```
        awk '/B03T3056/' merlin_out.03 > da_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > da_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > da_abs_09.lod

cd ../..
# Move into the next : Chr3/
cd Chr3

# Repeating all the steps done for Chr1

mkdir Present Absent
cd Present
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '$3 > 0' da.dat > tmp1.dat
    awk '$6 > 1' tmp1.dat > tmp2.dat
    awk '$117 < 2' tmp2.dat > tmp3.dat
    awk '$118 < 2' tmp3.dat > tmp4.dat
    awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
```

```
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ai.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == "$ped" da.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

        pull_pedigrees.pl da.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
```

```
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > da_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > da_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > da_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > da_pre_09.lod
```

```
        cd ../Absent
        cp ../../../../da.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        cp ../Present/pedin.dat pedin1.dat
        cat ./pedin1.dat ./da.dat | sort | uniq -u > pedin.dat
             awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

        ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_abs_01.lod
```

```
        awk '/B01T0558/' merlin_out.01 > da_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > da_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > da_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > da_abs_09.lod

cd ../..

# Move into the next: Chr5

cd Chr5

# Repeat steps

mkdir Present Absent
cd Present
```

```
        cp ../../../../da.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '$3 > 0' da.dat > tmp1.dat
        awk '$6 > 1' tmp1.dat > tmp2.dat
        awk '$165 < 2' tmp2.dat > tmp3.dat
        awk '$166 < 2' tmp3.dat > tmp4.dat
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file da.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == "$ped" da.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

        pull_pedigrees.pl da.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
```

```
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > da_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > da_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_pre_05.lod
```

```
            awk '/B05T4140/' merlin_out.05 > da_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_pre_09.lod

            awk '/B09T8333/' merlin_out.09 > da_pre_09.lod

      cd ../Absent
      cp ../../../../da.dat .
      cp ../../../../datain.dat .
      cp ../../../../map.dat .
      cp ../Present/pedin.dat pedin1.dat
      cat ./pedin1.dat ./da.dat | sort | uniq -u > pedin.dat
          awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
```

```
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_abs_01.lod

        awk '/B01T0558/' merlin_out.01 > da_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > da_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > da_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > da_abs_09.lod

cd ../..
```

```
# Move into the next : Chr9

cd Chr9

# Repeat steps

mkdir Present Absent
cd Present
        cp ../../../../da.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '$3 > 0' da.dat > tmp1.dat
        awk '$6 > 1' tmp1.dat > tmp2.dat
        awk '$191 < 2' tmp2.dat > tmp3.dat
        awk '$192 < 2' tmp3.dat > tmp4.dat
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file da.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" da.dat >> pedin.dat
# end
```

```
# or (2) code to call pull_pedigrees.pl to do the work:

      pull_pedigrees.pl da.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

      ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_pre_01.lod

            awk '/B01T0558/' merlin_out.01 > da_pre_01.lod
```

```
# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > da_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > da_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > da_pre_09.lod

    cd ../Absent
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Present/pedin.dat pedin1.dat
    cat ./pedin1.dat ./da.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
```

```
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_abs_01.lod

        awk '/B01T0558/' merlin_out.01 > da_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > da_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > da_abs_05.lod
```

```
# print the lines that contain the pattern B09T8333 in

# the file merlin_out.09 and write it to the file da_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > da_abs_09.lod

# Move up by three levels

cd ../../..

# Move into the DA_results directory and copy the results only

cd DA_results

mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1

cp ../../DA/Chr1/Present/pedlist1.dat .

cp ../../DA/Chr1/Present/da_pre* .

cp ../../DA/Chr1/Absent/pedlist2.dat .

cp ../../DA/Chr1/Absent/da_abs* .

cd ../Chr3

cp ../../DA/Chr3/Present/pedlist1.dat .

cp ../../DA/Chr3/Present/da_pre* .

cp ../../DA/Chr3/Absent/pedlist2.dat .

cp ../../DA/Chr3/Absent/da_abs* .

cd ../Chr5

cp ../../DA/Chr5/Present/pedlist1.dat .
```

```
cp ../../DA/Chr5/Present/da_pre* .

cp ../../DA/Chr5/Absent/pedlist2.dat .

cp ../../DA/Chr5/Absent/da_abs* .


cd ../Chr9

cp ../../DA/Chr9/Present/pedlist1.dat .

cp ../../DA/Chr9/Present/da_pre* .

cp ../../DA/Chr9/Absent/pedlist2.dat .

cp ../../DA/Chr9/Absent/da_abs* .


cd ../../



# Move into the next population KA and repeat the analyses.


cd KA

mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1

mkdir Present Absent

cd Present

    cp ../../../../ka.dat .

    cp ../../../../datain.dat .

    cp ../../../../map.dat .

    awk '$3 > 0' ka.dat > tmp1.dat

    awk '$6 > 1' tmp1.dat > tmp2.dat

    awk '$41 < 2' tmp2.dat > tmp3.dat

    awk '$42 < 2' tmp3.dat > tmp4.dat
```

```
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ka.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ka.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

        pull_pedigrees.pl ka.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
```

```
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_pre_09.lod
```

```
        cd ../Absent
        cp ../../../../ka.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .

# compare ka.dat with pedin.dat in the directory-present, then retain those
# lines that are not in pedin.dat and write it to a file called pedin.dat
        cp ../Present/pedin.dat pedin1.dat
        cat ./pedin1.dat ./ka.dat | sort | uniq -u > pedin.dat
            awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

        ./merlin.all.sh
```

```
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_abs_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_abs_09.lod

cd ../..
# Move into the next : Chr3/
cd Chr3

# Repeating all the steps done for Chr1
```

```
mkdir Present Absent
cd Present
        cp ../../../../ka.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '$3 > 0' ka.dat > tmp1.dat
        awk '$6 > 1' tmp1.dat > tmp2.dat
        awk '$117 < 2' tmp2.dat > tmp3.dat
        awk '$118 < 2' tmp3.dat > tmp4.dat
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ka.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ka.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

        pull_pedigrees.pl ka.dat pedlist1.dat pedin.dat

# running mega2
```

```
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_pre_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_pre_09.lod

    cd ../Absent
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Present/pedin.dat pedin1.dat
    cat ./pedin1.dat ./ka.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
```

```
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_abs_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_abs_09.lod
```

```
cd ../..

# Move into the next: Chr5

cd Chr5

# Repeat steps

mkdir Present Absent
cd Present
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '$3 > 0' ka.dat > tmp1.dat
    awk '$6 > 1' tmp1.dat > tmp2.dat
    awk '$165 < 2' tmp2.dat > tmp3.dat
    awk '$166 < 2' tmp3.dat > tmp4.dat
    awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
    rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ka.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
```

```
#   awk '$1 == '$ped" ka.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

      pull_pedigrees.pl ka.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

      ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_pre_01.lod
```

```
        awk '/B01T0558/' merlin_out.01 > ka_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_pre_09.lod

    cd ../Absent
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Present/pedin.dat pedin1.dat
    cat ./pedin1.dat ./ka.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
```

```
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_abs_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_abs_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_abs_05.lod
```

```
        awk '/B05T4140/' merlin_out.05 > ka_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_abs_09.lod

cd ../..

# Move into the next : Chr9

cd Chr9

# Repeat steps

mkdir Present Absent
cd Present
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '$3 > 0' ka.dat > tmp1.dat
    awk '$6 > 1' tmp1.dat > tmp2.dat
    awk '$191 < 2' tmp2.dat > tmp3.dat
    awk '$192 < 2' tmp3.dat > tmp4.dat
    awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
    rm tmp*
```

```
# if the number in the pedlist.dat matches the number in the first field
# of the file ka.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ka.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

      pull_pedigrees.pl ka.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
```

```
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_pre_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_pre_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_pre_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_pre_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_pre_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_pre_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_pre_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_pre_09.lod

    cd ../Absent
    cp ../../../../ka.dat .
```

```
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        cp ../Present/pedin.dat pedin1.dat
        cat ./pedin1.dat ./ka.dat | sort | uniq -u > pedin.dat
             awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

        ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_abs_01.lod

              awk '/B01T0558/' merlin_out.01 > ka_abs_01.lod
```

```
# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_abs_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_abs_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_abs_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_abs_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_abs_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_abs_09.lod

# Move up by three levels

cd ../../..

# Move into the KA_results directory and copy the results only

cd KA_results
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
cp ../../KA/Chr1/Present/pedlist1.dat .
cp ../../KA/Chr1/Present/ka_pre* .
cp ../../KA/Chr1/Absent/pedlist2.dat .
```

```
cp ../../KA/Chr1/Absent/ka_abs* .

cd ../Chr3
cp ../../KA/Chr3/Present/pedlist1.dat .
cp ../../KA/Chr3/Present/ka_pre* .
cp ../../KA/Chr3/Absent/pedlist2.dat .
cp ../../KA/Chr3/Absent/ka_abs* .

cd ../Chr5
cp ../../KA/Chr5/Present/pedlist1.dat .
cp ../../KA/Chr5/Present/ka_pre* .
cp ../../KA/Chr5/Absent/pedlist2.dat .
cp ../../KA/Chr5/Absent/ka_abs* .

cd ../Chr9
cp ../../KA/Chr9/Present/pedlist1.dat .
cp ../../KA/Chr9/Present/ka_pre* .
cp ../../KA/Chr9/Absent/pedlist2.dat .
cp ../../KA/Chr9/Absent/ka_abs* .

cd ../..

# Move into NY and repeat the analyses (this time using simwalk2)

cd NY
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
mkdir Present Absent
```

```
cd Present
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '$3 > 0' ny.dat > tmp1.dat
        awk '$6 > 1' tmp1.dat > tmp2.dat
        awk '$41 < 2' tmp2.dat > tmp3.dat
        awk '$42 < 2' tmp3.dat > tmp4.dat
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ny.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ny.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

        pull_pedigrees.pl ny.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
```

```
2
dat
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# copy the results into different files

cp STATS-01.ALL ny_pre_01.lod
cp STATS-03.ALL ny_pre_03.lod
cp STATS-05.ALL ny_pre_05.lod
cp STATS-09.ALL ny_pre_09.lod

cd ../Absent
    cp ../../../../ny.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
```

```
# compare ny.dat with pedin.dat in the directory-present, then retain those
# lines that are not in pedin.dat and write it to a file called pedin.dat
        cp ../Present/pedin.dat pedin1.dat
        cat ./pedin1.dat ./ny.dat | sort | uniq -u > pedin.dat
            awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_abs_01.lod
cp STATS-03.ALL ny_abs_03.lod
cp STATS-05.ALL ny_abs_05.lod
cp STATS-09.ALL ny_abs_09.lod
```

```
cd ../..
# Move into the next : Chr3/
cd Chr3

# Repeating all the steps done for Chr1

mkdir Present Absent
cd Present
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '$3 > 0' ny.dat > tmp1.dat
        awk '$6 > 1' tmp1.dat > tmp2.dat
        awk '$117 < 2' tmp2.dat > tmp3.dat
        awk '$118 < 2' tmp3.dat > tmp4.dat
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ny.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ny.dat >> pedin.dat
```

```
# end

# or (2) code to call pull_pedigrees.pl to do the work:

    pull_pedigrees.pl ny.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files

cp STATS-01.ALL ny_pre_01.lod
cp STATS-03.ALL ny_pre_03.lod
```

```
cp STATS-05.ALL ny_pre_05.lod

cp STATS-09.ALL ny_pre_09.lod


cd ../Absent

        cp ../../../../ny.dat .

        cp ../../../../datain.dat .

        cp ../../../../map.dat .

        cp ../Present/pedin.dat pedin1.dat

        cat ./pedin1.dat ./ny.dat | sort | uniq -u > pedin.dat

             awk '{print $1}' pedin.dat | sort -u > pedlist2.dat

# running mega2

mega2 << end_of_input

2

dat

0

1

3

y

4

1

8

0

end_of_input


mv map.dat map1.dat


./npl.all.sh
```

```
# Copy results into files
cp STATS-01.ALL ny_abs_01.lod
cp STATS-03.ALL ny_abs_03.lod
cp STATS-05.ALL ny_abs_05.lod
cp STATS-09.ALL ny_abs_09.lod

cd ../..

# Move into the next: Chr5

cd Chr5

# Repeat steps

mkdir Present Absent
cd Present
    cp ../../../../ny.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '$3 > 0' ny.dat > tmp1.dat
    awk '$6 > 1' tmp1.dat > tmp2.dat
    awk '$165 < 2' tmp2.dat > tmp3.dat
    awk '$166 < 2' tmp3.dat > tmp4.dat
    awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
    rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ny.dat, then retain those lines and write it to a
```

```
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ny.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

      pull_pedigrees.pl ny.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat
```

```
./npl.all.sh



# Copy results into files
cp STATS-01.ALL ny_pre_01.lod
cp STATS-03.ALL ny_pre_03.lod
cp STATS-05.ALL ny_pre_05.lod
cp STATS-09.ALL ny_pre_09.lod


cd ../Absent
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        cp ../Present/pedin.dat pedin1.dat
        cat ./pedin1.dat ./ny.dat | sort | uniq -u > pedin.dat
             awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
```

```
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_abs_01.lod
cp STATS-03.ALL ny_abs_03.lod
cp STATS-05.ALL ny_abs_05.lod
cp STATS-09.ALL ny_abs_09.lod

cd ../..

# Move into the next : Chr9

cd Chr9

# Repeat steps

mkdir Present Absent
cd Present
    cp ../../../../ny.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '$3 > 0' ny.dat > tmp1.dat
    awk '$6 > 1' tmp1.dat > tmp2.dat
```

```
        awk '$191 < 2' tmp2.dat > tmp3.dat
        awk '$192 < 2' tmp3.dat > tmp4.dat
        awk '{print $1}' tmp4.dat | sort -u > pedlist1.dat
        rm tmp*

# if the number in the pedlist.dat matches the number in the first field
# of the file ny.dat, then retain those lines and write it to a
# file called pedin.dat

# two ways to do this:
#  (1) code in this file to loop over pedigrees in pedlist.dat:

# foreach ped (`cat pedlist.dat`)
#   awk '$1 == '$ped" ny.dat >> pedin.dat
# end

# or (2) code to call pull_pedigrees.pl to do the work:

        pull_pedigrees.pl ny.dat pedlist1.dat pedin.dat

# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
```

```
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_pre_01.lod
cp STATS-03.ALL ny_pre_03.lod
cp STATS-05.ALL ny_pre_05.lod
cp STATS-09.ALL ny_pre_09.lod

cd ../Absent
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        cp ../Present/pedin.dat pedin1.dat
        cat ./pedin1.dat ./ny.dat | sort | uniq -u > pedin.dat
             awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
# running mega2
mega2 << end_of_input
2
dat
0
```

```
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_abs_01.lod
cp STATS-03.ALL ny_abs_03.lod
cp STATS-05.ALL ny_abs_05.lod
cp STATS-09.ALL ny_abs_09.lod

cd ../../..

# Move into the NY_results directory and copy the results only

cd NY_results
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
cp ../../NY/Chr1/Present/pedlist1.dat .
cp ../../NY/Chr1/Present/ny_pre* .
```

```
cp ../../NY/Chr1/Absent/pedlist2.dat .

cp ../../NY/Chr1/Absent/ny_abs* .


cd ../Chr3

cp ../../NY/Chr3/Present/pedlist1.dat .

cp ../../NY/Chr3/Present/ny_pre* .

cp ../../NY/Chr3/Absent/pedlist2.dat .

cp ../../NY/Chr3/Absent/ny_abs* .


cd ../Chr5

cp ../../NY/Chr5/Present/pedlist1.dat .

cp ../../NY/Chr5/Present/ny_pre* .

cp ../../NY/Chr5/Absent/pedlist2.dat .

cp ../../NY/Chr5/Absent/ny_abs* .


cd ../Chr9

cp ../../NY/Chr9/Present/pedlist1.dat .

cp ../../NY/Chr9/Present/ny_pre* .

cp ../../NY/Chr9/Absent/pedlist2.dat .

cp ../../NY/Chr9/Absent/ny_abs* .


cd ../../../


cd Redef

# REDEFINITION ANALYSES


mkdir AI DA KA NY

mkdir AI_results DA_results KA_results NY_results
```

```
cd AI
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
mkdir Homoz Heteroz
cd Homoz
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
# redefining phenotypes: affecteds that are homozygous for the '1' allele
    awk '{if($6 == 2 && $41 + $42 == 2) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
    cut -f 7- ai.dat > geno.out
    paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input
```

```
./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_homoz_09.lod


    cd ../Heteroz
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
```

```
        cp ../../../../map.dat .


# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
        awk '{if($6 == 2 && $41 + $42 == 3) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
        paste pheno.out geno.out > pedin.dat



# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input


        ./merlin.all.sh


# finding the peaks in the output files in merlin:
```

```
# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_heteroz_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_heteroz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_heteroz_09.lod


cd ../../Chr3

mkdir Homoz Heteroz
cd Homoz
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
```

```
        cp ../../../../map.dat .
        awk '{if($6 == 2 && $117 + $118 == 2) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
        paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

        ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_homoz_01.lod

                awk '/B01T0558/' merlin_out.01 > ai_homoz_01.lod
```

```
# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_homoz_09.lod


    cd ../Heteroz
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .

# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
    awk '{if($6 == 2 && $117 + $118 == 3) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
    cut -f 7- ai.dat > geno.out
    paste pheno.out geno.out > pedin.dat
```

```
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_heteroz_03.lod
```

```
        awk '/B03T3056/' merlin_out.03 > ai_heteroz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_heteroz_09.lod

cd ../../Chr5
mkdir Homoz Heteroz
cd Homoz
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '{if($6 == 2 && $165 + $166 == 4) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
    cut -f 7- ai.dat > geno.out
    paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
```

```
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_homoz_05.lod
```

```
# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_homoz_09.lod


    cd ../Heteroz
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .

# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
    awk '{if($6 == 2 && $165 + $166 == 3) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
    cut -f 7- ai.dat > geno.out
    paste pheno.out geno.out > pedin.dat


# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
```

```
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_heteroz_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_heteroz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_heteroz_09.lod
```

```
        awk '/B09T8333/' merlin_out.09 > ai_heteroz_09.lod


cd ../../Chr9
mkdir Homoz Heteroz
cd Homoz
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '{if($6 == 2 && $191 + $192 == 4) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
    cut -f 7- ai.dat > geno.out
    paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input
```

```
./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_homoz_09.lod


    cd ../Heteroz
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
```

```
        cp ../../../../map.dat .


# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
        awk '{if($6 == 2 && $191 + $192 == 3) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
        paste pheno.out geno.out > pedin.dat



# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input


        ./merlin.all.sh


# finding the peaks in the output files in merlin:
```

```
# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_heteroz_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_heteroz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_heteroz_09.lod


cd ../../../
# Move into the AI_results directory and copy the results only

cd AI_results
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
```

```
cp ../../AI/Chr1/Homoz/ai_homoz* .

cp ../../AI/Chr1/Heteroz/ai_heteroz* .


cd ../Chr3

cp ../../AI/Chr3/Homoz/ai_homoz* .

cp ../../AI/Chr3/Heteroz/ai_heteroz* .


cd ../Chr5

cp ../../AI/Chr5/Homoz/ai_homoz* .

cp ../../AI/Chr5/Heteroz/ai_heteroz* .


cd ../Chr9

cp ../../AI/Chr9/Homoz/ai_homoz* .

cp ../../AI/Chr9/Heteroz/ai_heteroz* .


cd ../../

# Move into the next population DA and repeat the analyses.


cd DA

mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1

mkdir Homoz Heteroz

cd Homoz

    cp ../../../../da.dat .

    cp ../../../../datain.dat .

    cp ../../../../map.dat .

    awk '{if($6 == 2 && $41 + $42 == 2) print $1, $2, $3, $4, $5, $6;\
```

```
        else print $1, $2, $3, $4, $5, "1"}' da.dat > pheno.out
    cut -f 7- da.dat > geno.out
    paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > da_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
```

```
# the file merlin_out.03 and write it to the file da_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > da_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > da_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > da_homoz_09.lod


    cd ../Heteroz
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .

# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
    awk '{if($6 == 2 && $41 + $42 == 3) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' da.dat > pheno.out
    cut -f 7- da.dat > geno.out
    paste pheno.out geno.out > pedin.dat
```

```
# running mega2

mega2 << end_of_input

2

dat

0

27

y

4

1

0

8

0

--npl --markerNames --bits 38

end_of_input


    ./merlin.all.sh


# finding the peaks in the output files in merlin:


# print the lines that contain the pattern B01T0558 in

# the file merlin_out.01 and write it to the file da_heteroz_01.lod


        awk '/B01T0558/' merlin_out.01 > da_heteroz_01.lod


# print the lines that contain the pattern B03T3056 in

# the file merlin_out.03 and write it to the file da_heteroz_03.lod


        awk '/B03T3056/' merlin_out.03 > da_heteroz_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > da_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > da_heteroz_09.lod


cd ../../Chr3

mkdir Homoz Heteroz
cd Homoz
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '{if($6 == 2 && $117 + $118 == 2) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' da.dat > pheno.out
    cut -f 7- da.dat > geno.out
    paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
```

```
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > da_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > da_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > da_homoz_05.lod
```

```
# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > da_homoz_09.lod


    cd ../Heteroz
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .

# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
    awk '{if($6 == 2 && $117 + $118 == 3) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' da.dat > pheno.out
    cut -f 7- da.dat > geno.out
    paste pheno.out geno.out > pedin.dat


# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
```

```
1

0

8

0

--npl --markerNames --bits 38

end_of_input


        ./merlin.all.sh


# finding the peaks in the output files in merlin:


# print the lines that contain the pattern B01T0558 in

# the file merlin_out.01 and write it to the file da_heteroz_01.lod


                awk '/B01T0558/' merlin_out.01 > da_heteroz_01.lod


# print the lines that contain the pattern B03T3056 in

# the file merlin_out.03 and write it to the file da_heteroz_03.lod


                awk '/B03T3056/' merlin_out.03 > da_heteroz_03.lod


# print the lines that contain the pattern B05T4140 in

# the file merlin_out.05 and write it to the file da_heteroz_05.lod


                awk '/B05T4140/' merlin_out.05 > da_heteroz_05.lod


# print the lines that contain the pattern B09T8333 in

# the file merlin_out.09 and write it to the file da_heteroz_09.lod
```

```
        awk '/B09T8333/' merlin_out.09 > da_heteroz_09.lod


cd ../../Chr5
mkdir Homoz Heteroz
cd Homoz
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '{if($6 == 2 && $165 + $166 == 4) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' da.dat > pheno.out
    cut -f 7- da.dat > geno.out
    paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input
```

```
./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > da_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > da_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > da_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > da_homoz_09.lod


    cd ../Heteroz
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
```

```
        cp ../../../../map.dat .


# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
        awk '{if($6 == 2 && $165 + $166 == 3) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' da.dat > pheno.out
        cut -f 7- da.dat > geno.out
        paste pheno.out geno.out > pedin.dat



# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input


        ./merlin.all.sh


# finding the peaks in the output files in merlin:
```

```
# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > da_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_heteroz_03.lod

        awk '/B03T3056/' merlin_out.03 > da_heteroz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > da_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > da_heteroz_09.lod

cd ../../Chr9
mkdir Homoz Heteroz
cd Homoz
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '{if($6 == 2 && $191 + $192 == 4) print $1, $2, $3, $4, $5, $6;\
```

```
        else print $1, $2, $3, $4, $5, "1"}' da.dat > pheno.out
    cut -f 7- da.dat > geno.out
    paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > da_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
```

```
# the file merlin_out.03 and write it to the file da_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > da_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > da_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > da_homoz_09.lod


    cd ../Heteroz
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .

# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
    awk '{if($6 == 2 && $191 + $192 == 3) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' da.dat > pheno.out
    cut -f 7- da.dat > geno.out
    paste pheno.out geno.out > pedin.dat
```

```
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_heteroz_01.lod

    awk '/B01T0558/' merlin_out.01 > da_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_heteroz_03.lod

    awk '/B03T3056/' merlin_out.03 > da_heteroz_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_heteroz_05.lod

    awk '/B05T4140/' merlin_out.05 > da_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_heteroz_09.lod

    awk '/B09T8333/' merlin_out.09 > da_heteroz_09.lod


# Move into DA_results and copy the results

cd ../../../DA_results
mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1
cp ../../DA/Chr1/Homoz/da_homoz* .
cp ../../DA/Chr1/Heteroz/da_heteroz* .

cd ../Chr3
cp ../../DA/Chr3/Homoz/da_homoz* .
cp ../../DA/Chr3/Heteroz/da_heteroz* .

cd ../Chr5
cp ../../DA/Chr5/Homoz/da_homoz* .
cp ../../DA/Chr5/Heteroz/da_heteroz* .
```

```
cd ../Chr9
cp ../../DA/Chr9/Homoz/da_homoz* .
cp ../../DA/Chr9/Heteroz/da_heteroz* .

cd ../../
# Move into KA and repeat the analyses

cd KA
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
mkdir Homoz Heteroz
cd Homoz
        cp ../../../../ka.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '{if($6 == 2 && $41 + $42 == 2) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ka.dat > pheno.out
        cut -f 7- ka.dat > geno.out
        paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
```

```
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_homoz_09.lod
```

```
        awk '/B09T8333/' merlin_out.09 > ka_homoz_09.lod


    cd ../Heteroz

    cp ../../../../ka.dat .

    cp ../../../../datain.dat .

    cp ../../../../map.dat .

# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
    awk '{if($6 == 2 && $41 + $42 == 3) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ka.dat > pheno.out
    cut -f 7- ka.dat > geno.out
    paste pheno.out geno.out > pedin.dat


# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
```

```
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_heteroz_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_heteroz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_heteroz_09.lod
```

```
cd ../../Chr3


mkdir Homoz Heteroz

cd Homoz

        cp ../../../..//ka.dat .

        cp ../../../../datain.dat .

        cp ../../../../map.dat .

        awk '{if($6 == 2 && $117 + $118 == 2) print $1, $2, $3, $4, $5, $6;\

                else print $1, $2, $3, $4, $5, "1"}' ka.dat > pheno.out

        cut -f 7- ka.dat > geno.out

        paste pheno.out geno.out > pedin.dat

# running mega2

mega2 << end_of_input

2

dat

0

27

y

4

1

0

8

0

--npl --markerNames --bits 38

end_of_input


        ./merlin.all.sh
```

```
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_homoz_09.lod


    cd ../Heteroz
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
```

```
# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
        awk '{if($6 == 2 && $117 + $118 == 3) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ka.dat > pheno.out
        cut -f 7- ka.dat > geno.out
        paste pheno.out geno.out > pedin.dat



# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

        ./merlin.all.sh


# finding the peaks in the output files in merlin:
```

```
# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_heteroz_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_heteroz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_heteroz_09.lod

cd ../../Chr5
mkdir Homoz Heteroz
cd Homoz
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '{if($6 == 2 && $165 + $166 == 4) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ka.dat > pheno.out
```

```
        cut -f 7- ka.dat > geno.out
        paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

        ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_homoz_01.lod

                awk '/B01T0558/' merlin_out.01 > ka_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_homoz_03.lod
```

```
        awk '/B03T3056/' merlin_out.03 > ka_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_homoz_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_homoz_09.lod


    cd ../Heteroz
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .

# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
    awk '{if($6 == 2 && $165 + $166 == 3) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ka.dat > pheno.out
    cut -f 7- ka.dat > geno.out
    paste pheno.out geno.out > pedin.dat


# running mega2
```

```
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_heteroz_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_heteroz_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_heteroz_09.lod

cd ../../Chr9
mkdir Homoz Heteroz
cd Homoz
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    awk '{if($6 == 2 && $191 + $192 == 4) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ka.dat > pheno.out
    cut -f 7- ka.dat > geno.out
    paste pheno.out geno.out > pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
```

```
1
0
8
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_homoz_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_homoz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_homoz_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_homoz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_homoz_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_homoz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_homoz_09.lod
```

```
        awk '/B09T8333/' merlin_out.09 > ka_homoz_09.lod


    cd ../Heteroz

    cp ../../../../ka.dat .

    cp ../../../../datain.dat .

    cp ../../../../map.dat .

# the next group of redefined phenotypes with the affecteds and
# heterozygotes as affecteds
    awk '{if($6 == 2 && $191 + $192 == 3) print $1, $2, $3, $4, $5, $6;\
        else print $1, $2, $3, $4, $5, "1"}' ka.dat > pheno.out
    cut -f 7- ka.dat > geno.out
    paste pheno.out geno.out > pedin.dat


# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
```

```
0
--npl --markerNames --bits 38
end_of_input

    ./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_heteroz_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_heteroz_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_heteroz_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_heteroz_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_heteroz_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_heteroz_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_heteroz_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_heteroz_09.lod
```

```
cd ../../../

# Move into the KA_results directory and copy the results

cd KA_results
mkdir Chr1 Chr3 Chr5 Chr9
cd Chr1
cp ../../KA/Chr1/Homoz/ka_homoz* .
cp ../../KA/Chr1/Heteroz/ka_heteroz* .

cd ../Chr3
cp ../../KA/Chr3/Homoz/ka_homoz* .
cp ../../KA/Chr3/Heteroz/ka_heteroz* .

cd ../Chr5
cp ../../KA/Chr5/Homoz/ka_homoz* .
cp ../../KA/Chr5/Heteroz/ka_heteroz* .

cd ../Chr9
cp ../../KA/Chr9/Homoz/ka_homoz* .
cp ../../KA/Chr9/Heteroz/ka_heteroz* .

cd ../..

# Move into NY and repeat the analyses (using simwalk2)

cd NY
mkdir Chr1 Chr3 Chr5 Chr9
```

```
cd Chr1

mkdir Homoz Heteroz

cd Homoz

        cp ../../../../ny.dat .

        cp ../../../../datain.dat .

        cp ../../../../map.dat .

        awk '{if($6 == 2 && $41 + $42 == 2) print $1, $2, $3, $4, $5, $6;\

                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out

        cut -f 7- ai.dat > geno.out

        paste pheno.out geno.out > pedin.dat


# running mega2

mega2 << end_of_input

2

dat

0

1

3

y

4

1

8

0

end_of_input


mv map.dat map1.dat


./npl.all.sh
```

```
# Copy results into files
cp STATS-01.ALL ny_homoz_01.lod
cp STATS-03.ALL ny_homoz_03.lod
cp STATS-05.ALL ny_homoz_05.lod
cp STATS-09.ALL ny_homoz_09.lod

cd ../Heteroz
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '{if($6 == 2 && $41 + $42 == 3) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
        paste pheno.out geno.out > pedin.dat


# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
```

```
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_heteroz_01.lod
cp STATS-03.ALL ny_heteroz_03.lod
cp STATS-05.ALL ny_heteroz_05.lod
cp STATS-09.ALL ny_heteroz_09.lod


cd ../..
# Move into the next : Chr3/
cd Chr3

# Repeating all the steps done for Chr1

mkdir Homoz Heteroz
cd Homoz
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '{if($6 == 2 && $117 + $118 == 2) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
```

```
paste pheno.out geno.out > pedin.dat


# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_homoz_01.lod
cp STATS-03.ALL ny_homoz_03.lod
cp STATS-05.ALL ny_homoz_05.lod
cp STATS-09.ALL ny_homoz_09.lod

cd ../Heteroz
```

```
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '{if($6 == 2 && $117 + $118 == 3) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
        paste pheno.out geno.out > pedin.dat


# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
```

```
cp STATS-01.ALL ny_heteroz_01.lod

cp STATS-03.ALL ny_heteroz_03.lod

cp STATS-05.ALL ny_heteroz_05.lod

cp STATS-09.ALL ny_heteroz_09.lod


cd ../..


# Move into the next: Chr5


cd Chr5


# Repeat steps


mkdir Homoz Heteroz

cd Homoz

        cp ../../../../ny.dat .

        cp ../../../../datain.dat .

        cp ../../../../map.dat .

        awk '{if($6 == 2 && $165 + $166 == 4) print $1, $2, $3, $4, $5, $6;\

                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out

        cut -f 7- ai.dat > geno.out

        paste pheno.out geno.out > pedin.dat



# running mega2

mega2 << end_of_input

2

dat
```

```
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_homoz_01.lod
cp STATS-03.ALL ny_homoz_03.lod
cp STATS-05.ALL ny_homoz_05.lod
cp STATS-09.ALL ny_homoz_09.lod

cd ../Heteroz
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '{if($6 == 2 && $165 + $166 == 3) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
        paste pheno.out geno.out > pedin.dat
```

```
# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_heteroz_01.lod
cp STATS-03.ALL ny_heteroz_03.lod
cp STATS-05.ALL ny_heteroz_05.lod
cp STATS-09.ALL ny_heteroz_09.lod

cd ../..
```

```
# Move into the next : Chr9

cd Chr9

# Repeat steps

mkdir Homoz Heteroz
cd Homoz
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '{if($6 == 2 && $191 + $192 == 4) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
        paste pheno.out geno.out > pedin.dat



# running mega2
mega2 << end_of_input
2
dat
0
1
3
y
4
1
8
```

```
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_homoz_01.lod
cp STATS-03.ALL ny_homoz_03.lod
cp STATS-05.ALL ny_homoz_05.lod
cp STATS-09.ALL ny_homoz_09.lod

cd ../Heteroz
        cp ../../../../ny.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        awk '{if($6 == 2 && $191 + $192 == 3) print $1, $2, $3, $4, $5, $6;\
                else print $1, $2, $3, $4, $5, "1"}' ai.dat > pheno.out
        cut -f 7- ai.dat > geno.out
        paste pheno.out geno.out > pedin.dat


# running mega2
mega2 << end_of_input
2
dat
0
```

```
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files
cp STATS-01.ALL ny_heteroz_01.lod
cp STATS-03.ALL ny_heteroz_03.lod
cp STATS-05.ALL ny_heteroz_05.lod
cp STATS-09.ALL ny_heteroz_09.lod

cd ../../../

# Move into NY_results and copy the results

cd NY_results
mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1
cp ../../NY/Chr1/Homoz/ny_homoz* .
```

```
cp ../../NY/Chr1/Heteroz/ny_heteroz* .

cd ../Chr3
cp ../../NY/Chr3/Homoz/ny_homoz* .
cp ../../NY/Chr3/Heteroz/ny_heteroz* .

cd ../Chr5
cp ../../NY/Chr5/Homoz/ny_homoz* .
cp ../../NY/Chr5/Heteroz/ny_heteroz* .

cd ../Chr9
cp ../../NY/Chr9/Homoz/ny_homoz* .
cp ../../NY/Chr9/Heteroz/ny_heteroz* .

cd ../../../

# Weighted analyses

cd Weight
mkdir AI DA KA NY
mkdir AI_results DA_results KA_results NY_results
# AI population - weighted analyses

cd AI
mkdir Chr1 Chr3 Chr5 Chr9 Genomescan
cd Genomescan
    cp ../../../ai.dat .
    cp ../../../datain.dat .
```

```
	cp ../../../map.dat .
	cp ai.dat pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38 --perFamily
end_of_input

# tweak the merlin shell scripts to add the line about merlin.lod
# extension changes
	echo "mv merlin.lod merlin.01.lod">>merlin.01.sh
	echo "mv merlin.lod merlin.03.lod">>merlin.03.sh
	echo "mv merlin.lod merlin.05.lod">>merlin.05.sh
	echo "mv merlin.lod merlin.09.lod">>merlin.09.sh
./merlin.all.sh

cd ../Chr1
mkdir Pos Neg
cd Pos
```

```
        cp ../../Genomescan/merlin.01.lod .
# print the lines with the pattern 'B01T0558'
        awk '/B01T0558/' merlin.01.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
        awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
        awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
        awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist.dat from ai.dat
        cp ../../../../ai.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        pull_pedigrees.pl ai.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input
```

```
./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_pos_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_pos_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_pos_09.lod

cd ../Neg
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Pos/pedin.dat pedin1.dat
```

```
        cat ./pedin1.dat ./ai.dat | sort | uniq -u > pedin.dat

            awk '{print $1}' pedin.dat | sort -u > pedlist2.dat

mega2 << end_of_input

2

dat

0

27

y

4

1

0

8

0

--npl --markerNames --bits 38

end_of_input


./merlin.all.sh

# finding the peaks in the output files in merlin:


# print the lines that contain the pattern B01T0558 in

# the file merlin_out.01 and write it to the file ai_neg_01.lod


              awk '/B01T0558/' merlin_out.01 > ai_neg_01.lod


# print the lines that contain the pattern B03T3056 in

# the file merlin_out.03 and write it to the file ai_neg_03.lod


              awk '/B03T3056/' merlin_out.03 > ai_neg_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_neg_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_neg_09.lod

cd ../../Chr3
mkdir Pos Neg
cd Pos
    cp ../../Genomescan/merlin.03.lod .
# print the lines with the pattern 'B03T3056'
    awk '/B03T3056/' merlin.03.lod > tmp1
# take merlin.03.lod and retain only the fields 1 & 8
    awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
    awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
    awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from ai.dat
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    pull_pedigrees.pl ai.dat pedlist1.dat pedin.dat
```

```
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_pos_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_pos_03.lod

# print the lines that contain the pattern B05T4140 in
```

```
# the file merlin_out.05 and write it to the file ai_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_pos_09.lod

cd ../Neg
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Pos/pedin.dat pedin1.dat
    cat ./pedin1.dat ./ai.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
```

```
end_of_input

./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in

# the file merlin_out.01 and write it to the file ai_neg_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_neg_01.lod

# print the lines that contain the pattern B03T3056 in

# the file merlin_out.03 and write it to the file ai_neg_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_neg_03.lod

# print the lines that contain the pattern B05T4140 in

# the file merlin_out.05 and write it to the file ai_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_neg_05.lod

# print the lines that contain the pattern B09T8333 in

# the file merlin_out.09 and write it to the file ai_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_neg_09.lod

cd ../../Chr5

mkdir Pos Neg

cd Pos
```

```
        cp ../../Genomescan/merlin.05.lod .
# print the lines with the pattern 'B05T4140'
        awk '/B05T4140/' merlin.05.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
        awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
        awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
        awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from ai.dat
        cp ../../../../ai.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        pull_pedigrees.pl ai.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input
```

```
./merlin.all.sh

# finding the peaks in the output files in merlin:


# print the lines that contain the pattern B01T0558 in

# the file merlin_out.01 and write it to the file ai_pos_01.lod


        awk '/B01T0558/' merlin_out.01 > ai_pos_01.lod


# print the lines that contain the pattern B03T3056 in

# the file merlin_out.03 and write it to the file ai_pos_03.lod


        awk '/B03T3056/' merlin_out.03 > ai_pos_03.lod


# print the lines that contain the pattern B05T4140 in

# the file merlin_out.05 and write it to the file ai_pos_05.lod


        awk '/B05T4140/' merlin_out.05 > ai_pos_05.lod


# print the lines that contain the pattern B09T8333 in

# the file merlin_out.09 and write it to the file ai_pos_09.lod


        awk '/B09T8333/' merlin_out.09 > ai_pos_09.lod


cd ../Neg

    cp ../../../../ai.dat .

    cp ../../../../datain.dat .

    cp ../../../../map.dat .

    cp ../Pos/pedin.dat pedin1.dat
```

```
cat ./pedin1.dat ./ai.dat | sort | uniq -u > pedin.dat

awk '{print $1}' pedin.dat | sort -u > pedlist2.dat

mega2 << end_of_input

2

dat

0

27

y

4

1

0

8

0

--npl --markerNames --bits 38

end_of_input


./merlin.all.sh

# finding the peaks in the output files in merlin:


# print the lines that contain the pattern B01T0558 in

# the file merlin_out.01 and write it to the file ai_neg_01.lod


awk '/B01T0558/' merlin_out.01 > ai_neg_01.lod


# print the lines that contain the pattern B03T3056 in

# the file merlin_out.03 and write it to the file ai_neg_03.lod


awk '/B03T3056/' merlin_out.03 > ai_neg_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ai_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_neg_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_neg_09.lod

cd ../../Chr9
mkdir Pos Neg
cd Pos
    cp ../../Genomescan/merlin.09.lod .
# print the lines with the pattern 'B09T8333'
    awk '/B05T8333/' merlin.09.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
    awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
    awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
    awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from ai.dat
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    pull_pedigrees.pl ai.dat pedlist1.dat pedin.dat
```

```
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ai_pos_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ai_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_pos_03.lod

# print the lines that contain the pattern B05T4140 in
```

```
# the file merlin_out.05 and write it to the file ai_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ai_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_pos_09.lod

cd ../Neg
    cp ../../../../ai.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Pos/pedin.dat pedin1.dat
    cat ./pedin1.dat ./ai.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
```

```
end_of_input

./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in

# the file merlin_out.01 and write it to the file ai_neg_01.lod

        awk '/B01T0558/' merlin_out.01 > ai_neg_01.lod

# print the lines that contain the pattern B03T3056 in

# the file merlin_out.03 and write it to the file ai_neg_03.lod

        awk '/B03T3056/' merlin_out.03 > ai_neg_03.lod

# print the lines that contain the pattern B05T4140 in

# the file merlin_out.05 and write it to the file ai_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > ai_neg_05.lod

# print the lines that contain the pattern B09T8333 in

# the file merlin_out.09 and write it to the file ai_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > ai_neg_09.lod

cd ../../../

# Move into AI_results and copy the files
```

```
cd AI_results

mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1

cp ../../AI/Chr1/Pos/pedlist1.dat .

cp ../../AI/Chr1/Pos/ai_pos* .

cp ../../AI/Chr1/Neg/pedlist2.dat .

cp ../../AI/Chr1/Neg/ai_neg* .


cd ../Chr3

cp ../../AI/Chr3/Pos/pedlist1.dat .

cp ../../AI/Chr3/Pos/ai_pos* .

cp ../../AI/Chr3/Neg/pedlist2.dat .

cp ../../AI/Chr3/Neg/ai_neg* .


cd ../Chr5

cp ../../AI/Chr5/Pos/pedlist1.dat .

cp ../../AI/Chr5/Pos/ai_pos* .

cp ../../AI/Chr5/Neg/pedlist2.dat .

cp ../../AI/Chr5/Neg/ai_neg* .


cd ../Chr9

cp ../../AI/Chr9/Pos/pedlist1.dat .

cp ../../AI/Chr9/Pos/ai_pos* .

cp ../../AI/Chr9/Neg/pedlist2.dat .

cp ../../AI/Chr9/Neg/ai_neg* .


cd ../../
```

```
# Move into the next population DA and repeat the analyses.


cd DA

mkdir Chr1 Chr3 Chr5 Chr9 Genomescan

cd Genomescan

        cp ../../../da.dat .

        cp ../../../datain.dat .

        cp ../../../map.dat .

        cp da.dat pedin.dat

# running mega2

mega2 << end_of_input

2

dat

0

27

y

4

1

0

8

0

--npl --markerNames --bits 38 --perFamily

end_of_input


# tweak the merlin shell scripts to add the line about merlin.lod

# extension changes

        echo "mv merlin.lod merlin.01.lod">>merlin.01.sh
```

```
        echo "mv merlin.lod merlin.03.lod">>merlin.03.sh
        echo "mv merlin.lod merlin.05.lod">>merlin.05.sh
        echo "mv merlin.lod merlin.09.lod">>merlin.09.sh
./merlin.all.sh

cd ../Chr1
mkdir Pos Neg
cd Pos
        cp ../../Genomescan/merlin.01.lod .
# print the lines with the pattern 'B01T0558'
        awk '/B01T0558/' merlin.01.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
        awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
        awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
        awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from da.dat
        cp ../../../../da.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        pull_pedigrees.pl da.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
```

```
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_pos_01.lod

        awk '/B01T0558/' merlin_out.01 > da_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > da_pos_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > da_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_pos_09.lod
```

```
            awk '/B09T8333/' merlin_out.09 > da_pos_09.lod


cd ../Neg
        cp ../../../../da.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        cp ../Pos/pedin.dat pedin1.dat
        cat ./pedin1.dat ./da.dat | sort | uniq -u > pedin.dat
            awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input


./merlin.all.sh
# finding the peaks in the output files in merlin:


# print the lines that contain the pattern B01T0558 in
```

```
# the file merlin_out.01 and write it to the file da_neg_01.lod

        awk '/B01T0558/' merlin_out.01 > da_neg_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_neg_03.lod

        awk '/B03T3056/' merlin_out.03 > da_neg_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > da_neg_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > da_neg_09.lod

cd ../../Chr3
mkdir Pos Neg
cd Pos
    cp ../../Genomescan/merlin.03.lod .
# print the lines with the pattern 'B03T3056'
    awk '/B03T3056/' merlin.03.lod > tmp1
# take merlin.03.lod and retain only the fields 1 & 8
    awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
```

```
        awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
        awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from da.dat
        cp ../../../../da.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        pull_pedigrees.pl da.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_pos_01.lod
```

```
        awk '/B01T0558/' merlin_out.01 > da_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > da_pos_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > da_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > da_pos_09.lod

cd ../Neg
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Pos/pedin.dat pedin1.dat
    cat ./pedin1.dat ./da.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
```

```
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_neg_01.lod

        awk '/B01T0558/' merlin_out.01 > da_neg_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_neg_03.lod

        awk '/B03T3056/' merlin_out.03 > da_neg_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > da_neg_05.lod
```

```
# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > da_neg_09.lod

cd ../../Chr5
mkdir Pos Neg
cd Pos
    cp ../../Genomescan/merlin.05.lod .
# print the lines with the pattern 'B05T4140'
    awk '/B05T4140/' merlin.05.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
    awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
    awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
    awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from da.dat
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    pull_pedigrees.pl da.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
```

```
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_pos_01.lod

        awk '/B01T0558/' merlin_out.01 > da_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > da_pos_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > da_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_pos_09.lod
```

```
            awk '/B09T8333/' merlin_out.09 > da_pos_09.lod


cd ../Neg
        cp ../../../../da.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        cp ../Pos/pedin.dat pedin1.dat
        cat ./pedin1.dat ./da.dat | sort | uniq -u > pedin.dat
            awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input


./merlin.all.sh
# finding the peaks in the output files in merlin:


# print the lines that contain the pattern B01T0558 in
```

```
# the file merlin_out.01 and write it to the file da_neg_01.lod

        awk '/B01T0558/' merlin_out.01 > da_neg_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_neg_03.lod

        awk '/B03T3056/' merlin_out.03 > da_neg_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > da_neg_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > da_neg_09.lod

cd ../../Chr9
mkdir Pos Neg
cd Pos
    cp ../../Genomescan/merlin.09.lod .
# print the lines with the pattern 'B09T8333'
    awk '/B05T8333/' merlin.09.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
    awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
```

```
        awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
        awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from da.dat
        cp ../../../../da.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        pull_pedigrees.pl da.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_pos_01.lod
```

```
        awk '/B01T0558/' merlin_out.01 > da_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > da_pos_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > da_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file da_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > da_pos_09.lod

cd ../Neg
    cp ../../../../da.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Pos/pedin.dat pedin1.dat
    cat ./pedin1.dat ./da.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
```

```
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file da_neg_01.lod

        awk '/B01T0558/' merlin_out.01 > da_neg_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file da_neg_03.lod

        awk '/B03T3056/' merlin_out.03 > da_neg_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file da_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > da_neg_05.lod
```

```
# print the lines that contain the pattern B09T8333 in

# the file merlin_out.09 and write it to the file da_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > da_neg_09.lod

cd ../../../

# Move into DA_results and copy the files

cd DA_results

mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1

cp ../../DA/Chr1/Pos/pedlist1.dat .

cp ../../DA/Chr1/Pos/da_pos* .

cp ../../DA/Chr1/Neg/pedlist2.dat .

cp ../../DA/Chr1/Neg/da_neg* .

cd ../Chr3

cp ../../DA/Chr3/Pos/pedlist1.dat .

cp ../../DA/Chr3/Pos/da_pos* .

cp ../../DA/Chr3/Neg/pedlist2.dat .

cp ../../DA/Chr3/Neg/da_neg* .

cd ../Chr5

cp ../../DA/Chr5/Pos/pedlist1.dat .

cp ../../DA/Chr5/Pos/da_pos* .

cp ../../DA/Chr5/Neg/pedlist2.dat .

cp ../../DA/Chr5/Neg/da_neg* .
```

```
cd ../Chr9
cp ../../DA/Chr9/Pos/pedlist1.dat .
cp ../../DA/Chr9/Pos/da_pos* .
cp ../../DA/Chr9/Neg/pedlist2.dat .
cp ../../DA/Chr9/Neg/da_neg* .

cd ../../

# Move into the next population KA and repeat the analyses.

cd KA
mkdir Chr1 Chr3 Chr5 Chr9 Genomescan
cd Genomescan
    cp ../../../ka.dat .
    cp ../../../datain.dat .
    cp ../../../map.dat .
    cp ka.dat pedin.dat
# running mega2
mega2 << end_of_input
2
dat
0
27
y
4
1
0
```

```
8
0
--npl --markerNames --bits 38 --perFamily
end_of_input

# tweak the merlin shell scripts to add the line about merlin.lod
# extension changes
        echo "mv merlin.lod merlin.01.lod">>merlin.01.sh
        echo "mv merlin.lod merlin.03.lod">>merlin.03.sh
        echo "mv merlin.lod merlin.05.lod">>merlin.05.sh
        echo "mv merlin.lod merlin.09.lod">>merlin.09.sh
./merlin.all.sh

cd ../Chr1
mkdir Pos Neg
cd Pos
        cp ../../Genomescan/merlin.01.lod .
# print the lines with the pattern 'B01T0558'
        awk '/B01T0558/' merlin.01.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
        awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
        awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
        awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from ka.dat
        cp ../../../../ka.dat .
        cp ../../../../datain.dat .
```

```
        cp ../../../../map.dat .
        pull_pedigrees.pl ka.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_pos_01.lod

            awk '/B01T0558/' merlin_out.01 > ka_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_pos_03.lod

            awk '/B03T3056/' merlin_out.03 > ka_pos_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_pos_09.lod

cd ../Neg
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Pos/pedin.dat pedin1.dat
    cat ./pedin1.dat ./ka.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
```

```
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_neg_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_neg_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_neg_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_neg_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_neg_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_neg_09.lod

cd ../../Chr3
```

```
mkdir Pos Neg
cd Pos
        cp ../../Genomescan/merlin.03.lod .
# print the lines with the pattern 'B03T3056'
        awk '/B03T3056/' merlin.03.lod > tmp1
# take merlin.03.lod and retain only the fields 1 & 8
        awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
        awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
        awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from ka.dat
        cp ../../../../ka.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        pull_pedigrees.pl ka.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
```

```
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_pos_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_pos_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_pos_09.lod

cd ../Neg
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
```

```
        cp ../../../../map.dat .
        cp ../Pos/pedin.dat pedin1.dat
        cat ./pedin1.dat ./ka.dat | sort | uniq -u > pedin.dat
             awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_neg_01.lod

              awk '/B01T0558/' merlin_out.01 > ka_neg_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_neg_03.lod
```

```
        awk '/B03T3056/' merlin_out.03 > ka_neg_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_neg_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_neg_09.lod

cd ../../Chr5
mkdir Pos Neg
cd Pos
    cp ../../Genomescan/merlin.05.lod .
# print the lines with the pattern 'B05T4140'
    awk '/B05T4140/' merlin.05.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
    awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
    awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
    awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from ka.dat
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
```

```
        cp ../../../../map.dat .
        pull_pedigrees.pl ka.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_pos_01.lod

            awk '/B01T0558/' merlin_out.01 > ka_pos_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_pos_03.lod

            awk '/B03T3056/' merlin_out.03 > ka_pos_03.lod
```

```
# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_pos_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_pos_09.lod

cd ../Neg
    cp ../../../../ka.dat .
    cp ../../../../datain.dat .
    cp ../../../../map.dat .
    cp ../Pos/pedin.dat pedin1.dat
    cat ./pedin1.dat ./ka.dat | sort | uniq -u > pedin.dat
        awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
```

```
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_neg_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_neg_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_neg_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_neg_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_neg_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_neg_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_neg_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_neg_09.lod

cd ../../Chr9
```

```
mkdir Pos Neg
cd Pos
        cp ../../Genomescan/merlin.09.lod .
# print the lines with the pattern 'B09T8333'
        awk '/B05T8333/' merlin.09.lod > tmp1
# take merlin.01.lod and retain only the fields 1 & 8
        awk '{print $1, $8}' tmp1 > tmp2
# retain lines if column 2 > 0 ---> tmp3
        awk '$2>0' tmp2 > tmp3
# print the first field (pedigree numbers) to pedlist.dat
        awk '{print $1}' tmp3 > pedlist1.dat
# choose the pedigrees based on pedlist1.dat from ka.dat
        cp ../../../../ka.dat .
        cp ../../../../datain.dat .
        cp ../../../../map.dat .
        pull_pedigrees.pl ka.dat pedlist1.dat pedin.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
```

```
end_of_input

./merlin.all.sh

# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in

# the file merlin_out.01 and write it to the file ka_pos_01.lod

        awk '/B01T0558/' merlin_out.01 > ka_pos_01.lod

# print the lines that contain the pattern B03T3056 in

# the file merlin_out.03 and write it to the file ka_pos_03.lod

        awk '/B03T3056/' merlin_out.03 > ka_pos_03.lod

# print the lines that contain the pattern B05T4140 in

# the file merlin_out.05 and write it to the file ka_pos_05.lod

        awk '/B05T4140/' merlin_out.05 > ka_pos_05.lod

# print the lines that contain the pattern B09T8333 in

# the file merlin_out.09 and write it to the file ka_pos_09.lod

        awk '/B09T8333/' merlin_out.09 > ka_pos_09.lod

cd ../Neg

    cp ../../../../ka.dat .

    cp ../../../../datain.dat .
```

```
        cp ../../../../map.dat .
        cp ../Pos/pedin.dat pedin1.dat
        cat ./pedin1.dat ./ka.dat | sort | uniq -u > pedin.dat
             awk '{print $1}' pedin.dat | sort -u > pedlist2.dat
mega2 << end_of_input
2
dat
0
27
y
4
1
0
8
0
--npl --markerNames --bits 38
end_of_input

./merlin.all.sh
# finding the peaks in the output files in merlin:

# print the lines that contain the pattern B01T0558 in
# the file merlin_out.01 and write it to the file ka_neg_01.lod

              awk '/B01T0558/' merlin_out.01 > ka_neg_01.lod

# print the lines that contain the pattern B03T3056 in
# the file merlin_out.03 and write it to the file ka_neg_03.lod
```

```
awk '/B03T3056/' merlin_out.03 > ka_neg_03.lod

# print the lines that contain the pattern B05T4140 in
# the file merlin_out.05 and write it to the file ka_neg_05.lod

awk '/B05T4140/' merlin_out.05 > ka_neg_05.lod

# print the lines that contain the pattern B09T8333 in
# the file merlin_out.09 and write it to the file ka_neg_09.lod

awk '/B09T8333/' merlin_out.09 > ka_neg_09.lod

cd ../../../

# Move into KA_results and copy the files/results

cd KA_results
mkdir Chr1 Chr3 Chr5 Chr9

cd Chr1
cp ../../KA/Chr1/Pos/pedlist1.dat .
cp ../../KA/Chr1/Pos/ka_pos* .
cp ../../KA/Chr1/Neg/pedlist2.dat .
cp ../../KA/Chr1/Neg/ka_neg* .

cd ../Chr3
cp ../../KA/Chr3/Pos/pedlist1.dat .
cp ../../KA/Chr3/Pos/ka_pos* .
```

```
cp ../../KA/Chr3/Neg/pedlist2.dat .

cp ../../KA/Chr3/Neg/ka_neg* .


cd ../Chr5

cp ../../KA/Chr5/Pos/pedlist1.dat .

cp ../../KA/Chr5/Pos/ka_pos* .

cp ../../KA/Chr5/Neg/pedlist2.dat .

cp ../../KA/Chr5/Neg/ka_neg* .


cd ../Chr9

cp ../../KA/Chr9/Pos/pedlist1.dat .

cp ../../KA/Chr9/Pos/ka_pos* .

cp ../../KA/Chr9/Neg/pedlist2.dat .

cp ../../KA/Chr9/Neg/ka_neg* .


cd ../../


# Move into the next population NY and repeat the analyses


cd NY

mkdir Chr1 Chr3 Chr5 Chr9 Genomescan

cd Genomescan

    cp ../../../ny.dat .

    cp ../../../datain.dat .

    cp ../../../map.dat .

    cp ny.dat pedin.dat

# running mega2

mega2 << end_of_input
```

```
2
dat
0
1
3
y
4
1
8
0
end_of_input

mv map.dat map1.dat

./npl.all.sh

# Copy results into files (these have to be analyzed in order to split the families)

cp STATS-01.ALL ny_01.lod
cp STATS-03.ALL ny_03.lod
cp STATS-05.ALL ny_05.lod
cp STATS-09.ALL ny_09.lod

# Move into NY_results and copy the files

cd ../../NY_results
cp ../NY/Genomescan/ny_0* .
cd ../../
```

APPENDIX B

SHELL SCRIPT FOR LOGISTIC REGRESSION ANALYSIS

```
#!/bin/tcsh -f # C-shell file name: logist.sh
#-----------------------------------------------
#       Input file names:
#                       ai.dat
#                       da.dat
#                       ka.dat
#                       ny.dat
#-----------------------------------------------

mkdir AI DA KA NY
mkdir AI_results DA_results KA_results NY_results
cd AI
mkdir Chr1_3 Chr1_5 Chr1_9 Chr3_5 Chr3_9 Chr5_9
cd Chr1_3
        cp ../../ai.dat .
        echo "1 200" > ai_1_3.dat
    awk '$3 == 0' ai.dat | awk '{print $6, $41, $42, $117, $118}' >> ai_1_3.dat
        lrmodel ai_1_3.dat ai_1_3.r1
```

```
cd ../Chr1_5
        cp ../../ai.dat .
    echo "1 200" > ai_1_5.dat
    awk '$3 == 0' ai.dat | awk '{print $6, $41, $42, $165, $166}' >> ai_1_5.dat
    lrmodel ai_1_5.dat ai_1_5.r1
cd ../Chr1_9
        cp ../../ai.dat .
    echo "1 200" > ai_1_9.dat
    awk '$3 == 0' ai.dat | awk '{print $6, $41, $42, $191, $192}' >> ai_1_9.dat
    lrmodel ai_1_9.dat ai_1_9.r1
cd ../Chr3_5
        cp ../../ai.dat .
    echo "1 200" > ai_3_5.dat
    awk '$3 == 0' ai.dat | awk '{print $6, $117, $118, $165, $166}' >> ai_3_5.dat
    lrmodel ai_3_5.dat ai_3_5.r1
cd ../Chr3_9
        cp ../../ai.dat .
    echo "1 200" > ai_3_9.dat
    awk '$3 == 0' ai.dat | awk '{print $6, $117, $118, $191, $192}' >> ai_3_9.dat
    lrmodel ai_3_9.dat ai_3_9.r1
cd ../Chr5_9
        cp ../../ai.dat .
    echo "1 200" > ai_5_9.dat
    awk '$3 == 0' ai.dat | awk '{print $6, $165, $166, $191, $192}' >> ai_5_9.dat
    lrmodel ai_5_9.dat ai_5_9.r1

# Move into AI_results and copy the files
```

```
cd ../../AI_results

cp ../AI/Chr1_3/ai_1_3.r1 .
cp ../AI/Chr1_5/ai_1_5.r1 .
cp ../AI/Chr1_9/ai_1_9.r1 .
cp ../AI/Chr3_5/ai_3_5.r1 .
cp ../AI/Chr3_9/ai_3_9.r1 .
cp ../AI/Chr5_9/ai_5_9.r1 .

cd ../DA
mkdir Chr1_3 Chr1_5 Chr1_9 Chr3_5 Chr3_9 Chr5_9
cd Chr1_3
    cp ../../da.dat .
    echo "1 200" > da_1_3.dat
    awk '$3 == 0' da.dat | awk '{print $6, $41, $42, $117, $118}' >> da_1_3.dat
    lrmodel da_1_3.dat da_1_3.r1
cd ../Chr1_5
    cp ../../da.dat .
    echo "1 200" > da_1_5.dat
    awk '$3 == 0' da.dat | awk '{print $6, $41, $42, $165, $166}' >> da_1_5.dat
    lrmodel da_1_5.dat da_1_5.r1
cd ../Chr1_9
    cp ../../da.dat .
    echo "1 200" > da_1_9.dat
    awk '$3 == 0' da.dat | awk '{print $6, $41, $42, $191, $192}' >> da_1_9.dat
    lrmodel da_1_9.dat da_1_9.r1
cd ../Chr3_5
    cp ../../da.dat .
```

```
        echo "1 200" > da_3_5.dat
        awk '$3 == 0' da.dat | awk '{print $6, $117, $118, $165, $166}' >> da_3_5.dat
        lrmodel da_3_5.dat da_3_5.r1
cd ../Chr3_9
        cp ../../da.dat .
        echo "1 200" > da_3_9.dat
        awk '$3 == 0' da.dat | awk '{print $6, $117, $118, $191, $192}' >> da_3_9.dat
        lrmodel da_3_9.dat da_3_9.r1
cd ../Chr5_9
        cp ../../da.dat .
        echo "1 200" > da_5_9.dat
        awk '$3 == 0' da.dat | awk '{print $6, $165, $166, $191, $192}' >> da_5_9.dat
        lrmodel da_5_9.dat da_5_9.r1

# Move into DA_results and copy the files

cd ../../DA_results
cp ../DA/Chr1_3/da_1_3.r1 .
cp ../DA/Chr1_5/da_1_5.r1 .
cp ../DA/Chr1_9/da_1_9.r1 .
cp ../DA/Chr3_5/da_3_5.r1 .
cp ../DA/Chr3_9/da_3_9.r1 .
cp ../DA/Chr5_9/da_5_9.r1 .

cd ../KA
mkdir Chr1_3 Chr1_5 Chr1_9 Chr3_5 Chr3_9 Chr5_9
cd Chr1_3
        cp ../../ka.dat .
```

```
        echo "1 200" > ka_1_3.dat
        awk '$3 == 0' ka.dat | awk '{print $6, $41, $42, $117, $118}' >> ka_1_3.dat
        lrmodel ka_1_3.dat ka_1_3.r1
cd ../Chr1_5
        cp ../../ka.dat .
        echo "1 200" > ka_1_5.dat
        awk '$3 == 0' ka.dat | awk '{print $6, $41, $42, $165, $166}' >> ka_1_5.dat
        lrmodel ka_1_5.dat ka_1_5.r1
cd ../Chr1_9
        cp ../../ka.dat .
        echo "1 200" > ka_1_9.dat
        awk '$3 == 0' ka.dat | awk '{print $6, $41, $42, $191, $192}' >> ka_1_9.dat
        lrmodel ka_1_9.dat ka_1_9.r1
cd ../Chr3_5
        cp ../../ka.dat .
        echo "1 200" > ka_3_5.dat
        awk '$3 == 0' ka.dat | awk '{print $6, $117, $118, $165, $166}' >> ka_3_5.dat
        lrmodel ka_3_5.dat ka_3_5.r1
cd ../Chr3_9
        cp ../../ka.dat .
        echo "1 200" > ka_3_9.dat
        awk '$3 == 0' ka.dat | awk '{print $6, $117, $118, $191, $192}' >> ka_3_9.dat
        lrmodel ka_3_9.dat ka_3_9.r1
cd ../Chr5_9
        cp ../../ka.dat .
        echo "1 200" > ka_5_9.dat
        awk '$3 == 0' ka.dat | awk '{print $6, $165, $166, $191, $192}' >> ka_5_9.dat
        lrmodel ka_5_9.dat ka_5_9.r1
```

```
# Move into KA_results and copy the files

cd ../../KA_results
cp ../KA/Chr1_3/ka_1_3.r1 .
cp ../KA/Chr1_5/ka_1_5.r1 .
cp ../KA/Chr1_9/ka_1_9.r1 .
cp ../KA/Chr3_5/ka_3_5.r1 .
cp ../KA/Chr3_9/ka_3_9.r1 .
cp ../KA/Chr5_9/ka_5_9.r1 .

cd ../NY
mkdir Chr1_3 Chr1_5 Chr1_9 Chr3_5 Chr3_9 Chr5_9
cd Chr1_3
    cp ../../ny.dat .
    echo "1 100" > ny_1_3.dat
    awk '$3 == 0' ny.dat | awk '{print $6, $41, $42, $117, $118}' >> ny_1_3.dat
    lrmodel ny_1_3.dat ny_1_3.r1
cd ../Chr1_5
    cp ../../ny.dat .
    echo "1 100" > ny_1_5.dat
    awk '$3 == 0' ny.dat | awk '{print $6, $41, $42, $165, $166}' >> ny_1_5.dat
    lrmodel ny_1_5.dat ny_1_5.r1
cd ../Chr1_9
    cp ../../ny.dat .
    echo "1 100" > ny_1_9.dat
    awk '$3 == 0' ny.dat | awk '{print $6, $41, $42, $191, $192}' >> ny_1_9.dat
    lrmodel ny_1_9.dat ny_1_9.r1
```

```
cd ../Chr3_5
        cp ../../ny.dat .
        echo "1 100" > ny_3_5.dat
        awk '$3 == 0' ny.dat | awk '{print $6, $117, $118, $165, $166}' >> ny_3_5.dat
        lrmodel ny_3_5.dat ny_3_5.r1
cd ../Chr3_9
        cp ../../ny.dat .
        echo "1 100" > ny_3_9.dat
        awk '$3 == 0' ny.dat | awk '{print $6, $117, $118, $191, $192}' >> ny_3_9.dat
        lrmodel ny_3_9.dat ny_3_9.r1
cd ../Chr5_9
        cp ../../ny.dat .
        echo "1 100" > ny_5_9.dat
        awk '$3 == 0' ny.dat | awk '{print $6, $165, $166, $191, $192}' >> ny_5_9.dat
        lrmodel ny_5_9.dat ny_5_9.r1

# Move into NY_results and copy the files

cd ../../NY_results
cp ../NY/Chr1_3/ny_1_3.r1 .
cp ../NY/Chr1_5/ny_1_5.r1 .
cp ../NY/Chr1_9/ny_1_9.r1 .
cp ../NY/Chr3_5/ny_3_5.r1 .
cp ../NY/Chr3_9/ny_3_9.r1 .
cp ../NY/Chr5_9/ny_5_9.r1 .

cd ..
```

BIBLIOGRAPHY

1. Abecasis GR, Cherny SS, Cookson WO, Cardon Lr. Merlin. Rapid analysis of dense genetic maps using sparse gene flow trees. Nat Genet 2002; 30(1):97-101.

2. Anderson TJ. The dangers of using single locus markers in parasite epidemiology: Ascaris as a case study. Trends in Parasitology 2001. Vol.17 No.4;183-188.

3. Badano JL, Katsanis N. Beyond Mendel: an evolving view of human genetic disease transmission. Nat Rev Genet. 2002 Oct;3(10):779-89. Review.

4. Burmeister M. Basic concepts in the study of diseases with complex genetics. Biol Psychiatry. 1999 Mar 1;45(5):522-32. Review.

5. Burton PR, Tobin MD, Hopper JL. Key concepts in genetic epidemiology. Lancet. 2005 Sep 10-16;366(9489):941-51. Review.

6. Cardon LR, Bell JI. Association study designs for complex diseases. Nat Rev Genet. 2001 Feb;2(2):91-9. Review.

7. Clayton D. A generalization of the transmission/disequilibrium test for uncertain-haplotype transmission. Am J Hum Genet. 1999 Oct;65(4):1170-7.

8. Clayton D, McKeigue PM. Epidemiological methods for studying genes and environmental factors in complex diseases. Lancet. 2001 Oct 20;358(9290):1356-60. Review.

9. Cordell HJ. Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans. Hum Mol Genet. 2002 Oct 1;11(20):2463-8. Review.

10. Cordell HJ, Clayton DG. Genetic association studies. Lancet. 2005 Sep 24-30;366(9491):1121-31.

11. Cox NJ, Frigge M, Nicolae DL, Concannon P, Hanis CL, Bell GI, Kong A. Loci on chromosomes 2 (NIDDM1) and 15 interact to increase susceptibility to diabetes in Mexican Americans. Nat Genet. 1999 Feb;21(2):213-5.

12. Daly MJ, Rioux JD, Schaffner SF, Hudson TJ, Lander ES. High-resolution haplotype structure in the human genome. Nat Genet. 2001 Oct;29(2):229-32.

13. Dawn Teare M, Barrett JH. Genetic linkage studies. Lancet. 2005 Sep 17-23;366(9490):1036-44. Review.

14. Dean M. Approaches to identify genes for complex human diseases: lessons from Mendelian disorders. Hum Mutat. 2003 Oct;22(4):261-74. Review.

15. Deperthes BD, Meheus A, O'Reilly K, Broutet N. Maternal and congenital syphilis programmes: case studies in Bolivia, Kenya and South Africa. Bull World Health Organ. 2004 Jun;82(6):410-6. Review.

16. Devlin B, Roeder K, Wasserman L. Analysis of multilocus models of association. Genet Epidemiol. 2003 Jul;25(1):36-47. Review.

17. Dipple KM, McCabe ER. Modifier genes convert "simple" Mendelian disorders to complex traits. Mol Genet Metab. 2000 Sep-Oct;71(1-2):43-50. Review.

18. Duerr RH. The genetics of inflammatory bowel disease. Gastroenterol Clin North Am. 2002 Mar;31(1):63-76. Review.

19. Duerr RH, Barmada MM, Zhang L, Achkar JP, Cho JH, Hanauer SB, Brant SR, Bayless TM, Baldassano RN, Weeks DE. Evidence for an inflammatory bowel disease locus on chromosome 3p26: linkage, transmission/disequilibrium and partitioning of linkage. Hum Mol Genet. 2002 Oct 1;11(21):2599-606.

20. Duerr RH. Update on the genetics of inflammatory bowel disease. J Clin Gastroenterol. 2003 Nov-Dec;37(5):358-67. Review.

21. Falk CT, Rubinstein P. Haplotype relative risks: an easy reliable way to construct a proper control sample for risk calculations. Ann Hum Genet. 1987 Jul;51(Pt 3):227-33.

22. Fanous AH, Kendler KS. Genetic heterogeneity, modifier genes, and quantitative phenotypes in psychiatric illness: searching for a framework. Mol Psychiatry. 2005 Jan;10(1):6-13. Review.

23. Faraone SV, Tsuang MT. Heterogeneity and the genetics of bipolar disorder. Am J Med Genet C Semin Med Genet. 2003 Nov 15;123(1):1-9. Review.

24. Frankel WN, Schork NJ. Who's afraid of epistasis? Nat Genet. 1996 Dec;14(4):371-3.

25. Gabriel SB, Schaffner SF, Nguyen H, Moore JM, Roy J, Blumenstiel B, Higgins J, DeFelice M, Lochner A, Faggart M, Liu-Cordero SN, Rotimi C, Adeyemo A, Cooper R, Ward R, Lander ES, Daly MJ, Altshuler D. The structure of haplotype blocks in the human genome. Science. 2002 Jun 21;296(5576):2225-9.

26. Gauderman WJ, Duncan CT. The role of interacting determinants in the localization of genes. Advances in Genetics. 2001 Vol 42:393-409.

27. Giallourakis C, Stoll M, Miller K, Hampe J, Lander ES, Daly MJ, Schreiber S, Rioux JD. IBD5 is a general risk factor for inflammatory bowel disease: replication of association with Crohn disease and identification of a novel association with ulcerative colitis. Am J Hum Genet. 2003 Jul;73(1):205-11.

28. Glidden DV, Liang KY, Chiu YF, Pulver AE. Multipoint affected sibpair linkage methods for localizing susceptibility genes of complex diseases. Genet Epidemiol. 2003 Feb;24(2):107-17.

29. Gurling H, Smyth C, Kalsi G, Moloney E, Rifkin L, O'Neill J, Murphy P, Curtis D, Petursson H, Brynjolfsson J. Linkage findings in bipolar disorder. Nat Genet. 1995 May;10(1):8-9.

30. Hampe J, Wienker T, Nurnberg P, Schreiber S. Mapping genes for polygenic disorders: considerations for study design in the complex trait of inflammatory bowel disease. Hum Hered. 2000 Mar-Apr;50(2):91-101.

31. Hampe J, Cuthbert A, Croucher PJ, Mirza MM, Mascheretti S, Fisher S, Frenzel H, King K, Hasselmeyer A, MacPherson AJ, Bridger S, van Deventer S, Forbes A, Nikolaus S, Lennard-Jones JE, Foelsch UR, Krawczak M, Lewis C, Schreiber S, Mathew CG. Association between insertion mutation in NOD2 gene and Crohn's disease in German and British populations. Lancet. 2001 Jun 16;357(9272):1925-8.

32. Hamshere ML, Williams NM, Norton N, Williams H, Cardno AG, Zammit S, Jones LA, Murphy KC, Sanders RD, McCarthy G, Gray MY, Jones G, Holmans P, O'Donovan MC, Owen MJ, Craddock N. Genome wide significant linkage in schizophrenia conditioning on occurrence of depressive episodes. J Med Genet. 2006 Jul;43(7):563-7.

33. Hastbacka J, de la Chapelle A, Kaitila I, Sistonen P, Weaver A, Lander E. Linkage disequilibrium mapping in isolated founder populations: diastrophic dysplasia in Finland. Nat Genet. 1992 Nov;2(3):204-11.

34. Hattersley AT, McCarthy MI. What makes a good genetic association study? Lancet. 2005 Oct 8;366(9493):1315-23.

35. Hinds D, Risch N. The ASPEX package: affected sib-pair mapping, version 2.3. ftp://lahmed.stanford.edu/pub/aspex.

36. Hoffmann R, Valencia A. Protein interaction: same network, different hubs. Trends Genet. 2003 Dec;19(12):681-3. Review.

37. Hoffman S, Ober C. Present status on the genetic studies of asthma. Curr Opin Immunol. 2002 Dec;14(6):709-17. Review.

38. Hoh J, Ott J. Genetic dissection of diseases: design and methods. Curr Opin Genet Dev. 2004 Jun;14(3):229-32. Review.

39. Hopper JL, Bishop DT, Easton DF. Population-based family studies in genetic epidemiology. Lancet. 2005 Oct 15-21;366(9494):1397-406. Review.

40. Hugot JP, Laurent-Puig P, Gower-Rousseau C, Caillat-Zucman S, Beaugerie L, Dupas JL, Van Gossum A, Bonait-Pellie C, Cortot A, Thomas G. Linkage analyses of chromosome 6 loci, including HLA, in familial aggregations of Crohn disease. G.E.T.A.I.D. Am J Med Genet. 1994 Aug 15;52(2):207-13.

41. Hugot JP, Chamaillard M, Zouali H, Lesage S, Cezard JP, Belaiche J, Almer S, Tysk C, O'Morain CA, Gassull M, Binder V, Finkel Y, Cortot A, Modigliani R, Laurent-Puig P, Gower-Rousseau C, Macry J, Colombel JF, Sahbatou M, Thomas G. Association of NOD2 leucine-rich repeat variants with susceptibility to Crohn's disease. Nature. 2001 May 31;411(6837):599-603.

42. Johnson GC, Esposito L, Barratt BJ, Smith AN, Heward J, Di Genova G, Ueda H, Cordell HJ, Eaves IA, Dudbridge F, Twells RC, Payne F, Hughes W, Nutland S, Stevens H, Carr P, Tuomilehto-Wolf E, Tuomilehto J, Gough SC, Clayton DG, Todd JA. Haplotype tagging for the identification of common disease genes. Nat Genet. 2001 Oct;29(2):233-7.

43. Jones HB, Faham M. Evidence and implications for multiplicative interactions among loci predisposing to human common disease. Hum Hered. 2005;59(3):176-84.

44. Kaabi B, Gelernter J, Woods SW, Goddard A, Page GP, Elston RC. Genome scan for loci predisposing to anxiety disorders using a novel multivariate approach: strong evidence for a chromosome 4 risk locus. Am J Hum Genet. 2006 Apr;78(4):543-53.

45. Kennedy JL, Farrer LA, Andreasen NC, Mayeux R, St George-Hyslop P. The genetics of adult-onset neuropsychiatric disease: complexities and conundra? Science. 2003 Oct 31;302(5646):822-6. Review.

46. Kirkpatrick M, Johnson T, Barton N. General models of multilocus evolution. Genetics. 2002 Aug;161(4):1727-50. Review.

47. Kong A, Cox NJ. Allele-sharing models: LOD scores and accurate linkage tests. Am J Hum Genet. 1997 Nov;61(5):1179-88.

48. Kuida S, Beier DR. Genetic localization of interacting modifiers affecting severity in a murine model of polycystic kidney disease. Genome Res. 2000 Jan;10(1):49-54.

49. Lakatos PL, Fischer S, Lakatos L, Gal I, Papp J. Current concept on the pathogenesis of inflammatory bowel disease-crosstalk between genetic and microbial factors: pathogenic bacteria and altered bacterial sensing or changes in mucosal integrity take "toll" ? World J Gastroenterol. 2006 Mar 28;12(12):1829-41. Review.

50. Lander ES, Schork NJ. Genetic dissection of complex traits. Science. 1994 Sep 30;265(5181):2037-48. Review.

51. Lander E, Kruglyak L. Genetic dissection of complex traits: guidelines for interpreting and reporting linkage results. Nat Genet. 1995 Nov;11(3):241-7.

52. Leboyer M, Henry C, Paillere-Martinot ML, Bellivier F. Age at onset in bipolar affective disorders: a review. Bipolar Disord. 2005 Apr;7(2):111-8. Review.

53. MacGregor AJ, Snieder H, Schork NJ, Spector TD. Twins. Novel uses to study complex traits and genetic diseases. Trends Genet. 2000 Mar;16(3):131-4. Review.

54. Maniatis N, Collins A, Gibson J, Zhang W, Tapper W, Morton NE. Positional cloning by linkage disequilibrium. Am J Hum Genet. 2004 May;74(5):846-55.

55. Marchini J, Donnelly P, Cardon LR. Genome-wide strategies for detecting multiple loci that influence complex diseases. Nat Genet. 2005 Apr;37(4):413-7.

56. Martin ER, Monks SA, Warren LL, Kaplan NL. A test for linkage and association in general pedigrees: the pedigree disequilibrium test. Am J Hum Genet. 2000 Jul;67(1):146-54.

57. Martinez FD. Complexities of the genetics of asthma. Am J Respir Crit Care Med. 1997 Oct;156(4 Pt 2):S117-22. Review.

58. McInnis MG, Dick DM, Willour VL, Avramopoulos D, MacKinnon DF, Simpson SG, Potash JB, Edenberg HJ, Bowman ES, McMahon FJ, Smiley C, Chellis JL, Huo Y, Diggs T, Meyer ET, Miller M, Matteini AT, Rau NL, DePaulo JR, Gershon ES, Badner JA, Rice JP, Goate AM, Detera-Wadleigh SD, Nurnberger JI, Reich T, Zandi PP, Foroud TM. Genome-wide scan and conditional analysis in bipolar disorder: evidence for genomic interaction in the National Institute of Mental Health genetics initiative bipolar pedigrees. Biol Psychiatry. 2003 Dec 1;54(11):1265-73.

59. Meng Z, Zaykin DV, Xu CF, Wagner M, Ehm MG. Selection of genetic markers for association analyses, using linkage disequilibrium and haplotypes. Am J Hum Genet. 2003 Jul;73(1):115-30.

60. Mitchell AA, Cutler DJ, Chakravarti A. Undetected genotyping errors cause apparent overtransmission of common alleles in the transmission/disequilibrium test. Am J Hum Genet. 2003 Mar;72(3):598-610.

61. Moore JH. A global view of epistasis. Nat Genet. 2005 Jan;37(1):13-4.

62. Morris MC, Evans DA, Bienias JL, Tangney CC, Bennett DA, Wilson RS, Aggarwal N, Schneider J. Consumption of fish and n-3 fatty acids and risk of incident Alzheimer disease. Arch Neurol. 2003 Jul;60(7):940-6.

63. Motsinger AA, Lee SL, Mellick G, Ritchie MD. GPNN: power studies and applications of a neural network method for detecting gene-gene interactions in studies of human disease. BMC Bioinformatics. 2006 Jan 25;7:39.

64. North BV, Curtis D, Sham PC. Application of logistic regression to case-control association studies involving two causative loci. Hum Hered. 2005;59(2):79-87.

65. Palmer LJ, Cardon LR. Shaking the tree: mapping complex disease genes with linkage disequilibrium. Lancet. 2005 Oct 1;366(9492):1223-34.

66. Parkes M, Barmada MM, Satsangi J, Weeks DE, Jewell DP, Duerr RH. The IBD2 locus shows linkage heterogeneity between ulcerative colitis and Crohn disease. Am J Hum Genet. 2000 Dec;67(6):1605-10.

67. Patil N, Berno AJ, Hinds DA, Barrett WA, Doshi JM, Hacker CR, Kautzer CR, Lee DH, Marjoribanks C, McDonough DP, Nguyen BT, Norris MC, Sheehan JB, Shen N, Stern D, Stokowski RP, Thomas DJ, Trulson MO, Vyas KR, Frazer KA, Fodor SP, Cox DR. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. Science. 2001 Nov 23;294(5547):1719-23.

68. Pauls D. Genetics of childhood disorders: IV. Linkage analysis. J Am Acad Child Adolesc Psychiatry. 1999 Jul;38(7):932-4. Review.

69. Peeters M, Cortot A, Vermeire S, Colombel JF. Familial and sporadic inflammatory bowel disease: different entities? Inflamm Bowel Dis. 2000 Nov;6(4):314-20. Review.

70. Phillips TJ, Belknap JK. Complex-trait genetics: emergence of multivariate strategies. Nat Rev Neurosci. 2002 Jun;3(6):478-85. Review.

71. Pierik M, Yang H, Barmada MM, Cavanaugh JA, Annese V, Brant SR, Cho JH, Duerr RH, Hugot JP, McGovern DP, Paavola-Sakki P, Radford-Smith GL, Pavli P, Silverberg MS, Schreiber S, Taylor KD, Vlietinck R; IBD International Genetics Consortium. The IBD international genetics consortium provides further evidence for linkage to IBD4 and shows gene-environment interaction. Inflamm Bowel Dis. 2005 Jan;11(1):1-7.

72. Rannala B, Reeve JP. High-resolution multipoint linkage-disequilibrium mapping in the context of a human genome sequence. Am J Hum Genet. 2001 Jul;69(1):159-78.

73. Reich T, Edenberg HJ, Goate A, Williams JT, Rice JP, Van Eerdewegh P, Foroud T, Hesselbrock V, Schuckit MA, Bucholz K, Porjesz B, Li TK, Conneally PM, Nurnberger JI Jr, Tischfield JA, Crowe RR, Cloninger CR, Wu W, Shears S, Carr K, Crose C, Willig C, Begleiter H. Genome-wide search for genes affecting the risk for alcohol dependence. Am J Med Genet. 1998 May 8;81(3):207-15.

74. Ritchie MD, White BC, Parker JS, Hahn LW, Moore JH. Optimization of neural network architecture using genetic programming improves detection and modeling of gene-gene interactions in studies of human diseases. BMC Bioinformatics. 2003 Jul 7;4:28.

75. Roses AD. Genetic associations. Lancet. 1998 Mar 21;351(9106):916.

76. Russell RK, Wilson DC, Satsangi J. Unravelling the complex genetics of inflammatory bowel disease. Arch Dis Child. 2004 Jul;89(7):598-603. Review.

77. Schulze TG, McMahon FJ. Genetic linkage and association studies in bipolar affective disorder: a time for optimism. Am J Med Genet C Semin Med Genet. 2003 Nov 15;123(1):36-47. Review.

78. Schulze TG, Buervenich S, Badner JA, Steele CJ, Detera-Wadleigh SD, Dick D, Foroud T, Cox NJ, MacKinnon DF, Potash JB, Berrettini WH, Byerley W, Coryell W, DePaulo JR Jr, Gershon ES, Kelsoe JR, McInnis MG, Murphy DL, Reich T, Scheftner W, Nurnberger JI Jr, McMahon FJ. Loci on chromosomes 6q and 6p interact to increase susceptibility to bipolar affective disorder in the national institute of mental health genetics initiative pedigrees. Biol Psychiatry. 2004 Jul 1;56(1):18-23.

79. Scriver CR, Waters PJ. Monogenic traits are not simple: lessons from phenylketonuria. Trends Genet. 1999 Jul;15(7):267-72. Review.

80. Shaw SH, Hampe J, White R, Mathew CG, Curran ME, Schreiber S. Stratification by CARD15 variant genotype in a genome-wide search for inflammatory bowel disease susceptibility loci. Hum Genet. 2003 Nov;113(6):514-21.

81. Sillanpaa MJ, Auranen K. Replication in genetic studies of complex traits. Ann Hum Genet. 2004 Nov;68(Pt 6):646-57. Review.

82. Slager SL, Schaid DJ, Wang L, Thibodeau SN. Candidate-gene association studies with pedigree data: controlling for environmental covariates. Genet Epidemiol. 2003 May;24(4):273-83.

83. Smithies O. Many little things: one geneticist's view of complex diseases. Nat Rev Genet. 2005 May;6(5):419-25. Review.

84. Smyth C, Kalsi G, Curtis D, Brynjolfsson J, O'Neill J, Rifkin L, Moloney E, Murphy P, Petursson H, Gurling H. Two-locus admixture linkage analysis of bipolar and unipolar affective disorder supports the presence of susceptibility loci on chromosomes 11p15 and 21q22. Genomics. 1997 Feb 1;39(3):271-8.

85. Spielman RS, McGinnis RE, Ewens WJ. Transmission test for linkage disequilibrium: the insulin gene region and insulin-dependent diabetes mellitus (IDDM). Am J Hum Genet. 1993 Mar;52(3):506-16.

86. Sullivan PF, Neale BM, Neale MC, van den Oord E, Kendler KS. Multipoint and single point non-parametric linkage analysis with imperfect data. Am J Med Genet B Neuropsychiatr Genet. 2003 Aug 15;121(1):89-94.

87. Suresh R, Ambrose N, Roe C, Pluzhnikov A, Wittke-Thompson JK, Ng MC, Wu X, Cook EH, Lundstrom C, Garsten M, Ezrati R, Yairi E, Cox NJ. New complexities in the genetics of stuttering: significant sex-specific linkage signals. Am J Hum Genet. 2006 Apr;78(4):554-63.

88. Tekin M, Duman T, Bogoclu G, Incesulu A, Cin S, Akar N. Moderate hearing loss and pseudodominant inheritance due to L90P/35delG mutations in the GJB2 (connexin 26) gene. Genet Couns. 2003;14(4):379-86. Review.

89. Terwilliger JD, Ott J. A haplotype-based 'haplotype relative risk' approach to detecting allelic associations. Hum Hered. 1992;42(6):337-46.

90. Thomas A, Camp NJ. Graphical modeling of the joint distribution of alleles at associated loci. Am J Hum Genet. 2004 Jun;74(6):1088-101.

91. Thomson G. Mapping disease genes: family-based association studies. Am J Hum Genet. 1995 Aug;57(2):487-98.

92. Thomson G. Analysis of complex human genetic traits: an ordered-notation method and new tests for mode of inheritance. Am J Hum Genet. 1995 Aug;57(2):474-86.

93. Vieland VJ, Huang J. Two-locus heterogeneity cannot be distinguished from two-locus epistasis on the basis of affected-sib-pair data. Am J Hum Genet. 2003 Aug;73(2):223-32.

94. Vineis P, Veglia F, Anttila S, Benhamou S, Clapper ML, Dolzan V, Ryberg D, Hirvonen A, Kremers P, Le Marchand L, Pastorelli R, Rannug A, Romkes M, Schoket B, Strange RC, Garte S, Taioli E. CYP1A1, GSTM1 and GSTT1 polymorphisms and lung cancer: a pooled analysis of gene-gene interactions. Biomarkers. 2004 May-Jun;9(3):298-305.

95. Wang Q, Chen Q, Zhao K, Wang L, Wang L, Traboulsi EI. Update on the molecular genetics of retinitis pigmentosa. Ophthalmic Genet. 2001 Sep;22(3):133-54. Review.

96. Wu R, Lin M. Functional mapping - how to map and study the genetic architecture of dynamic complex traits. Nat Rev Genet. 2006 Mar;7(3):229-37. Review.

97. Xu J, Meyers DA, Ober C, Blumenthal MN, Mellen B, Barnes KC, King RA, Lester LA, Howard TD, Solway J, Langefeld CD, Beaty TH, Rich SS, Bleecker ER, Cox NJ; Collaborative Study on the Genetics of Asthma. Genomewide screen and identification of gene-gene interactions for asthma-susceptibility loci in three U.S. populations: collaborative study on the genetics of asthma. Am J Hum Genet. 2001 Jun;68(6):1437-46.

98. Zhao LP, Li SS, Khalid N A method for the assessment of disease associations with single-nucleotide polymorphism haplotypes and environmental variables in case-control studies. Am J Hum Genet. 2003 May;72(5):1231-50.

99. Zhao J, Jin L, Xiong M. Test for interaction between two unlinked loci. Am J Hum Genet. 2006 Nov;79(5):831-45.

100. Zondervan KT, Cardon LR. The complex interplay among factors that influence allelic association. Nat Rev Genet. 2004 Feb;5(2):89-100. Review.