MAPPING GENES FOR QUANTITATIVE TRAITS USING SELECTED SAMPLES OF SIBLING PAIRS

by

Jin Peng Szatkiewicz

BS, Peking University, 1996

MA, Wayne State University, 2000

Submitted to the Graduate Faculty of

Graduate School of Public Health in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2004

UNIVERSITY OF PITTSBURGH

GRADUATE SCHOOL OF PUBLIC HEALTH

This dissertation was presented

by

Jin Peng Szatkiewicz

It was defended on

June 23, 2004

and approved by

Lisa Weissfeld, Ph.D., Professor, Department of Biostatistics, Graduate School of Public Health, University of Pittsburgh

Chien-Cheng (George) Tseng, Sc.D., Assistant Professor, Departments of Biostatistics and Human Genetics, Graduate School of Public Health, University of Pittsburgh

Daniel E. Weeks, Ph.D., Professor, Departments of Human Genetics and Biostatistics, Graduate School of Public Health, University of Pittsburgh

Eleanor Feingold, Ph.D., Associate Professor, Departments of Human Genetics and Biostatistics, Graduate School of Public Health, University of Pittsburgh
Dissertation Director

MAPPING GENES FOR QUANTITATIVE TRAITS USING SELECTED SAMPLES OF SIBLING PAIRS

Jin Peng Szatkiewicz, Ph.D.

University of Pittsburgh, 2004

One of the most important research areas in human genetics is the effort to map genes associated with complex diseases such as cancer, heart disease, and diabetes. The public health relevance of these kinds of work is that gene mapping will bring an understanding of genetic risk and protective factors, and a description of the interaction between environment and genetic variation. In the last ten years there has been a dramatic increase in the number of studies seeking to map genes for quantitative traits. This has caused an explosion of new work on statistical methods for human quantitative trait locus (QTL) mapping. However, little of that work has dealt with selected samples, which are more common than population samples for human studies. This dissertation focuses on sibling pairs and considers the most common types of selected sampling. I surveyed most QTL mapping methods in the literature to evaluate which are appropriate for selected samples, and also developed new statistics for selected samples. Using simulation and analytical approaches, I identified the most powerful statistics for each type of sampling considered. I then compared various sampling designs using the best statistic for each and gave guidelines for choosing appropriate and powerful designs under different scenarios.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

PREFACE

To my parents and teachers

1. INTRODUCTION

One of the most important research areas in human genetics is the effort to map genes associated with complex diseases such as cancer, heart disease, and diabetes. Gene mapping will bring an understanding of genetic risk and protective factors, and a description of the interaction between environment and genetic variation. This dissertation deals with linkage analysis methods for detecting loci that influence quantitative traits in humans, focusing on the use of selected sibling pairs. In this chapter, I shall introduce necessary genetics background, review some most important issues in quantitative trait linkage analysis, identify the major gaps in knowledge that this study seeks to address, and finally outline the roadmap of this dissertation.

1.1. GENETICS BACKGROUND

The human genome consists of 23 homologous pairs of chromosomes containing tens of thousands of genes. Of each pair, one is inherited from the father and the other from the mother. Genetic differences between two people exist because each site can be occupied by one of several variant genes, known as *alleles*. The two alleles at the same position, or *locus*, along the pair of chromosomes constitute a person's *genotype* at that locus.

The term *phenotype* denotes an observable characteristic or trait. By a *quantitative trait*, we mean any human trait that is measured on a numerical scale, although it most commonly refers to the traits that are measured continuously or nearly so. Examples include adult human

height, body mass index, blood pressure etc. Typically we refer to the loci that influence quantitative traits in humans as quantitative trait loci (QTLs). The alternative to a quantitative trait is generally described as a "qualitative" or "disease" trait, which usually in human studies means a binary variable. Human geneticists have traditionally focused on binary disease traits, but in the last ten years there has been a dramatic increase in the number of studies seeking to map genes for quantitative traits.

Gametes (sperm or eggs) are formed in a process called meiosis. At the end of meiosis, the chromosome pairs are split so that each gamete carries a single set of 23 chromosomes. During the first stage of meiosis, the chromosome strands pair up and may exchange sections of genetic material. This random shuffling of genetic material, termed *recombination*, is a source of genetic diversity. Two genes on a chromosome that are close together tend to be transmitted to the same gamete; they are said to be tightly linked. Two genes that are far apart, or on different chromosomes, are transmitted to the same or different gametes with equal probability; they are unlinked.

Linkage methods for mapping genes use family data and try to detect co-segregation of traits and shared genetic material within families. In linkage analysis, we study the correlation between trait values and one or more genetic markers. A *genetic maker* is any locus on the chromosome at which there is a variation. Transmission of markers from parents to offspring is observable, and the chromosomal location of genetic markers is known. Advances in genetics in the past two decades have led to the availability of increasingly dense sets of genetic markers.

The underlying principle of all the statistical methods for linkage analysis is that people who have similar traits should have higher than expected levels of sharing of genetic material near the genes that influence those traits. That is, at a locus unrelated to the trait the amount of

genetic sharing between family members should be determined (randomly) purely as a function of family relationship, and should have nothing to do with trait values. But at a locus that affects the trait, there should be greater sharing among people who have similar trait values and less sharing among people with dissimilar trait values. Because the sharing is driven by historical recombination events, the effects of increased sharing at the trait locus will also be seen at nearby markers.

The genetic sharing among family members referred to above is called *identity by descent* (IBD), which is defined as the sharing of an allele inherited from a common ancestor. This is distinguished from *identity by state* (IBS), which simply means having the same allele. Alleles shared IBS can be IBD or not. Any two unrelated people share no alleles IBD. Two siblings can share 0, 1 or 2 alleles IBD with probabilities ¼, ½, and ¼ respectively. And most other types of relatives share 0 or 1 alleles IBD. To do genetic mapping, the *actual* IBD sharing in the pedigree is estimated at each locus on the genome based on marker data. Because we have a probability model (or models) for recombination along the chromosome, estimates of IBD sharing at each locus can be improved by incorporating marker information from the entire chromosome, which is called *multipoint analysis*. Note that some of the theory derived assuming perfect IBD information (i.e. that the marker is infinitely polymorphic) only approximately true for estimated IBD information.

1.2. ISSUES IN HUMAN QTL LINKAGE ANALYSIS

1.2.1. Trait Models

Most QTL mapping methods are derived under some particular trait model, although the robustness of the methods to deviations from the assumed models varies. The simplest

commonly-used model is the single major gene model. The trait value, X, is written as $X = \mu + g + e$, where μ is the overall mean, g is the deviation from the mean due to the genotype at the major locus, and e is the environmental or residual variation. The environmental variation has mean zero and variance σ_e^2, is independently chosen for all individuals, and is usually modeled as being normally distributed. This model assumes that there is no interaction between the effects of genotype and environment. If we additionally assume that the genotype and environment are uncorrelated, we can decompose the overall trait variance as $\sigma^2 = \sigma_g^2 + \sigma_e^2$, where σ_g^2 is the genetic variance at the major locus and can be further decomposed into additive and dominance components, σ_a^2 and σ_d^2. In addition to the normality assumption for the random environmental component, many QTL mapping methods (see chapter 2 for details) additionally assume that the entire population trait distribution is approximately normal, even though a model with a major gene and a normal environmental component does not lead to an overall normal distribution.

Sometimes other effects are added to the model, for example, *polygenic effects* (effects of many small-effect genes unlinked to the major genes) and effects of shared environment among family members. With sibling pairs, a residual correlation between the pairs can be used to describe the sum of polygenic and common environmental components.

1.2.2. Pedigree Types

QTL mapping can be accomplished with almost any kind of family data, from nuclear families with or without parents to larger extended pedigrees involving as many as thousands of people. While it is common to do genetic mapping using sibling pairs, larger sibships and extended pedigrees are generally viewed to possess more statistical power for equal genotyping effort

under most genetic models. However, the mathematical and computational issues involved in using larger pedigrees are much more difficult. This dissertation focuses on sibling pairs, but many of the general results are applicable to larger sibships as well.

1.2.3. Sampling designs

There are a number of different sampling designs conventionally used in human QTL mapping. A *population sample* is a sample of families collected in a way that is not dependent on the trait values. On the other hand, in a *selected sample*, families are recruited based on the trait values of one (*single-proband* ascertainment) or more (*multiple-proband* ascertainment) members. In many cases, single-proband ascertainment is simpler and more natural than population sampling, because the probands with extreme trait values are already enrolled in other studies or are listed in registries. Sibling pairs in studies that use multiple-proband ascertainment are usually discordant and/or concordant pairs. In a discordant pair, one sibling has high trait value, and the other has low trait value; in a concordant pair, both siblings have high trait values or both have low trait values. One of the best-known examples is the extreme discordant sibling pair (EDSP) design by Risch and Zhang (1995), in which one samples discordant pairs with one sibling in the top 10% of the trait distribution and the other in the bottom 10%. There are several variations on EDSP sampling, including the moderately discordant sibling pairs (MDSP) design by Forrest and Feingold (2000), which uses a less stringent selection threshold (e.g. 35%); and the extreme discordant and concordant (EDAC) design by Gu et al. (1996), which combines extreme concordant pairs with extreme discordant pairs. Furthermore, affected sibling pairs already collected for linkage studies of binary traits have been used to map QTLs. Affected sibling pairs can be considered concordant pairs for any quantitative traits associated with the disease they were originally collected to study (e.g. glucose and insulin levels for diabetes). Most recently,

Purcell et al (2001) proposed an "*optimal sampling*" strategy that precisely ascertains the most informative pairs in the population. Figure 1 illustrates some examples of different sampling designs for one of the trait models I considered (i.e. model 1 of table 7 on page 72.)

To illustrate the range of sampling schemes that appear in the literature, table 1 lists every QTL linkage paper that appeared in the *American Journal of Human Genetics* from January of 2002 through March of 2004. Of the 31 studies listed in the table, only 7 used population sampling. Another 9 studies used single-proband samples and the remaining 15 studies used different types of multiple-proband samples. The popularity of selected sampling has to do with its great power for QTL mapping (Carey and Williamson 1991, Risch and Zhang 1995).

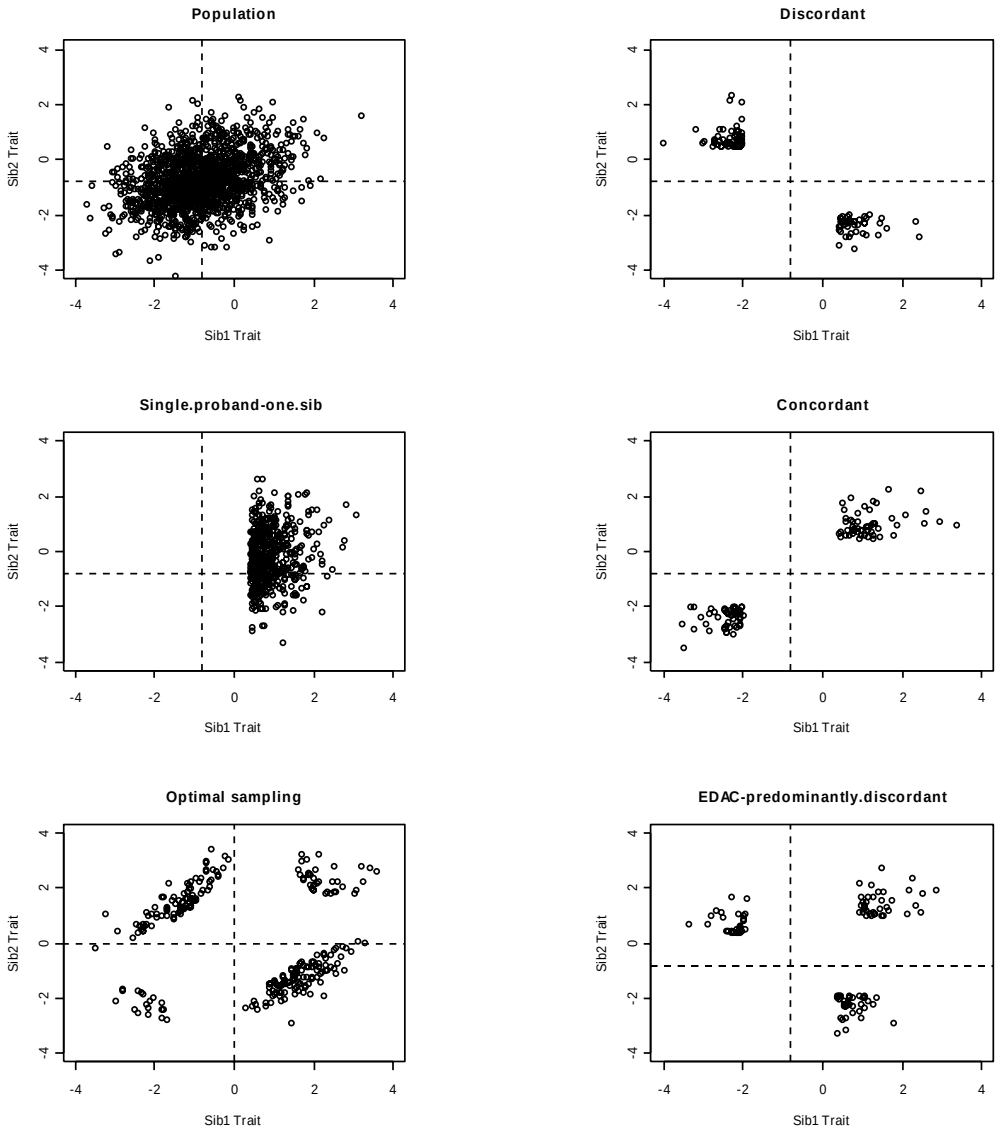

Figure 1 Bivariate scatter plots of various simulated samples from one trait model

Table 1 QTL Mapping Studies in the American Journal of Human Genetics, 2002 – 2004

First author	Trait	Type of pedigree	Type of ascertainment	Analysis method
			multiple-proband	
van Asselt	menopausal age	sib pairs	EDAC	H-E, adjusted VC
Fullerton	neuroticism	sib pairs	EDAC	IBD, Visscher&Hopper
Wilson	bone mineral density	sibships	EDAC	MAPMAKER/SIBS
Li	age of onset in HD	sib pairs	ASP with HD	VC
Dong	BMI	sib pairs	ASP with extreme obesity	H-E, VC
Fisher	ADHD symptom count	sib pairs	ASP with ADHD	H-E
Francks	relative hand skill	sib pairs	ASP with dyslexia	VC
O'Brien	language impairment	sib pairs	ASP with SLI	H-E
Alarcon	language traits	sibships	ASP with autism	MAPMAKER/SIBS
Wiltshire	stature	sibships	ASP with type 2 diabetes	VC
DeStefano	age of onset in Parkinson's	relative pairs	ASP with Parkinson's	VC
Slager	tumor aggressiveness	sibships	multiple cancer cases	HE
Wu	body mass index	sibships	mixed type multi-proband	VC
Pajukanta	HDL cholesterol	extended	complex multi-proband	binary trait linkage
Garner	fetal hemoglobin expression	very large	complex multi-proband	likelihood inference
			single-proband	
Kaplan	dyslexia phenotypes	sib pairs	proband with dyslexia	H-E, Defries-Fullker
Stein	speech and reading	sibships	proband with speech disorder	H-E
Arya	body mass index	extended	proband with type 2 diabetes	VC
Palmer	sleep apnea-related traits	extended	proband with apnea	VC
Silverman	spirometric phenotypes	extended	proband with early-onset COPD	VC
Soria	factor XII plasma levels	extended	proband with thrombophilia	VC
Deng	obesity	extended	proband with low bone density	VC
Xu	adult height	extended	proband with asthma	VC
Feitosa	body mass index	extended	proband with CHD	VC
			population	
Econs	bone mineral density	sib pairs	population	MAPMAKER/SIBS
Atwood	body mass index	sibships	population	VC
Lin	serum bilirubin	relative pairs	population	VC
Fox	carotid artery thickness	extended	population	VC
Feitosa	body mass index	extended	population	VC
Hall	T and B lymphocyte levels	extended	population	H-E
Abney	fasting serum-insulin	extended	founder population	homozygosity

1.2.4. Statistical methods

As of a few years ago, the most common analysis methods for QTL linkage using sibling pairs were Haseman-Elston regression (Haseman and Elston 1972) and maximum likelihood variance components analysis (e.g. Amos 1994, Almasy and Blangero 1998). The Haseman-Elston method regresses the squared differences in the sibling's trait values on their estimated identity-by-decent (IBD) sharing at a marker. If the marker is linked to the trait, high IBD sharing should be associated with a small difference in trait values, and the regression slope should be negative. Thus linkage can be tested with a regression t-test. The method was derived under the assumption of a population sample of sibling pairs with normally-distributed trait values, but the regression framework makes it quite robust to selected sampling and to non-Gaussian trait distributions.Variance components analysis, by contrast, relies on estimating variance components parameters under a Gaussian trait model. It has substantially higher power than Haseman-Elston regression under ideal conditions, but it can fail dramatically when the normality assumption is violated either by nonnormality of the trait distribution or by selected sampling (Feingold 2001, Feingold 2002).

Recently, there has been an explosion of new methods that aim to equal the power of variance components while retaining the robustness of Haseman-Elston regression. One set is regression-based statistics, essentially improvements on the original Haseman-Elston method. New regression-based methods have been developed by Drigalenko (1998), Elston et al. (2000), Xu et al. (2000), Forrest (2001), Visscher and Hopper (2001) and Sham and Purcell (2001). The whole idea of these methods is to do two separate regressions, one on squared trait differences and one on squared trait sums, and then to form a weighted average of the two estimates of

regression slopes. The other set of new methods is *score statistics*, based on the derivative of the usual variance components likelihood. The new score statistics are more computationally convenient than variance components, and they can be constructed in such a way as to be robust as well. The three primary papers on these methods are Tang and Siegmund (2001), Putter et al. (2002), and Wang and Huang (2002a). The score statistics proposed in these papers are very similar to each other, but have minor differences in how they parameterize the likelihood and how they alter the statistic to make it robust. Note that the above regression methods are primarily limited to sibling pairs; in case of score statistics they are naturally extended to larger pedigrees. Most recently, Sham et al. (2002) developed a robust regression-based method particularly for extended pedigrees; the basic idea is to reverse the original Haseman-Elston paradigm, and regress the IBD sharing on an appropriate function of the trait values. The best of these new methods are quite powerful and also potentially robust; however it is not clear how they compare to each other when they are applied to different types of selected sample under a variety of trait models.

There is a specialized set of methods to deal with a particular type of selected samples of sibling pairs - the discordant and concordant sibling pairs. These pairs possess a property that makes them critically different from more typical samples – they have a distorted IBD sharing distribution at markers linked to the trait. A population sample of sibling pairs is expected to share half of their alleles IBD at any locus, regardless of whether that locus is linked to the trait being studied. The same is true if the pairs are sampled on the basis of a single individual with an extreme trait value. But whenever a sibling pair is selected based on the phenotypes of both individuals, such as a discordant pair or a concordant pair, the IBD-sharing near the trait is lower or higher than the expected value (1/2) under null hypothesis. If the selection is extreme, this

change in the IBD sharing can be extreme. Risch and Zhang (1995) proposed collecting extreme discordant sibling pairs (EDSP) and tested linkage using the statistic, the average estimated IBD sharing for the pairs to detect the distorted IBD-sharing. This is the same statistic used for mapping binary traits using affected sibling pairs. Further extensions of this method adapted it to include concordant sibling pairs as well (e.g. Gu et al. 1996). However, these IBD-sharing statistics are limited in that they only look at the IBD sharing, ignoring the actual trait values. Consequently, the pairs must be very extreme so that the IBD sharing distribution is distorted enough to achieve high power to detect linkage.

Forrest and Feingold (2000) made the critical observation that IBD-sharing statistics and most of the regression-based statistics are independent under both the null and alternative hypotheses, and so may be combined to give a more powerful test of linkage. They proposed a composite statistic for discordant sibling pairs, which is a weighted sum of the IBD-sharing statistic and the Haseman-Elston (1972) regression t-statistic. They showed that their composite statistic is particularly useful for moderately discordant pairs and they suggested that it should be applicable to other multiple-proband sibling pair samples as well.

Based on the theory outline in Forrest and Feingold (2000), we categorize various QTL mapping statistics into three groups, based on the linkage information they use. The first group is *IBD-sharing statistics*; these statistics detect the distorted IBD-sharing distribution in discordant or concordant sibling pairs; however, they do not use trait value information at all. The second group is the *correlation-based statistics*; these statistics detect correlation between IBD-sharing and some measure of the similarity of trait values, but they do not use marginal IBD-sharing information. This group includes the majority of the regression methods and the variance components methods. The third group is the *combination statistics*; they combine linkage

information from both IBD-sharing and correlation, e.g. Forrest and Feingold's composite statistic. Note that score statistics can be constructed to be either correlation-based or combination. A complete detailed description of statistical methods will be introduced in chapter 2.

1.3. QTL MAPPING WITH SELECTED SIBLING PAIRS

1.3.1. Selected sampling and QTL mapping statistics

It has been amply demonstrated that phenotypically selected sibling pairs are generally more powerful than population samples of sibling pairs for linkage analysis of a quantitative trait locus (QTL) (e.g., Carey and Williamson 1991; Eaves and Meyer 1994; Risch and Zhang 1995). But whenever using any kind of selected samples, it is important to remember that selected sampling may introduce substantial departure from normality. In this case, the population distribution may be close to normal, but distribution of the ascertained sample will not be. It is well established that the variance components method loses significant power when normality assumption is violated (e.g. Allison et al. 1999, Sham et al. 2000, Tang and Siegmund 2001). Of the newer statistical methods, most were also explicitly or implicitly developed under the assumption of population sampling from a normally-distributed trait. In general, it is not yet clear which existing methods are appropriate and powerful for selected samples.

The new combination statistics deserve additional comments. Although developed for discordant sibling pairs, Forrest and Feingold's composite statistic demonstrated the potential of combination statistics for analyzing sibling pairs selected based on any multiple-proband criterion. Several other new statistics (e.g. Tang and Siegmund 2000) are potentially combination statistics, but they were not necessarily derived for selected samples. It is worthwhile to carefully

investigate the theory behind different combination statistics, to construct new powerful combination statistics for multiple-proband samples of various types, and to explore novel design possibilities enabled by combination statistics.

1.3.2. Literature comparing sampling designs

For sibling pairs, there has been a fair amount of literature comparing different sampling schemes. For example, Carey and Williamson (1991) compared single-proband sampling to population sampling. Risch and Zhang (1995) compared multiple-proband sampling to population sampling, and to the single-proband ascertainment schemes. Other papers that have dealt with similar issues include Zhang and Risch (1996), Gu et al. (1996,1997), Gu and Rao (1997b), Todorov et al.(1997), Rochberg et al. (1997), Rogus et al. (1997), Zhao et al. (1997), Allison et al. (1998), and Dolan and Boomsma (1998). I will not discuss this literature in depth, because its reliance on older statistics means that most of the results no longer hold when the more powerful new statistics are used instead. In addition, none of these studies was necessarily done with the best statistics for each selection scheme.

In all of the literature cited above, EDSP and EDAC designs were shown in theory to be quite powerful. However, such designs are only occasionally used in practice because it is very difficult to find extremely discordant pairs (typically one sibling in the top 10% of the trait distribution and one in the bottom 10%). For example, Fullerton et al. (2003) screened 20,427 independent sibships to get a final dataset of 182 discordant and 379 concordant pairs for neuroticism. Several authors (Gu and Rao 1997b, Forrest and Feingold 2000) have proposed using less restrictive thresholds and sampling “moderately” discordant pairs in order to achieve a balance between screening and genotyping efforts. However, it is not entirely clear when extreme sampling is advantageous, and when moderate sampling is better.

1.3.3. This dissertation

Thus we are left with a number of open questions about QTL mapping with selected samples of sibling pairs, which this dissertation seeks to explore. I considered population samples, single-proband samples and various types of multiple-proband samples of sibling pairs, and I took two constituent steps to understand (1) what is the best statistic for each type of sibling pair sample, and (2) what kinds of sampling are advantageous in what situations.

First I sought to know what the best statistic is for each type of sibling pair sample. I surveyed most of the exisitng QTL statistics and developed our own variants. I performed simulation to evaluate their type I error and power for population samples and for selected samples under a variety of trait models, and I examined the robustness of these methods to non-normality and to misspecification of trait parameters. In addition, I proved the independence between IBD-sharing statistics and correlation-based statistics, and revealed the relationship between different combination statistics that helps in suggesting directions for new powerful methods.

Next I sought to understand what kinds of sampling are advantageous in what situations. I compared different sampling schemes using the best method for each that was identified in the previous step, and studied how moderate sampling fits into the continuum given the new statistics available.

I have also developed easy-to-use, well-documented software that simulates various types of sibling pair samples, and that makes many of the new QTL statistics available to people conducting gene-mapping studies.

The organization of this dissertation is as follows: Chapter 2 provides an overall description and theoretical comparison of various QTL mapping statistics. Chapter 3 describes software. Chapter 4 compares statistics for population samples and single-proband samples, and several variants of the score statistics are also proposed. In chapter 5, I compare statistic for both extreme discordant sibling pairs and moderately discordant sibling pairs. In chapter 6, I propose a new robust discordant pair statistic that is better than all existing methods. In chapter 7, I first develop and compare statistics for EDAC sibling pair samples, and then compare different sampling designs that use sibling pairs. Chapter 8 discusses the additional study of statistics for QTL mapping with concordant and affected sibling pair samples. Finally, chapter 9 summarizes the results developed in this dissertation and discusses some open questions.

The work in chapters 4 and 5 were published in the *American Journal of Human Genetics* (T.Cuenco et al. 2003, Szatkiewicz et al. 2003). I have obtained copyright permission from the journal (The Chicargo Press) in order to reuse those two articles. Chapters 6 and 7 are based on our submitted manuscripts.

2. STATISTICAL METHODS

In this chapter I first summarize the theoretical foundation that is essential for understanding which statistics are appropriate for what kinds of samples; I then provide a detailed description and theoretical comparison of all of the statistics we considered.

2.1. THEORY

Forrest and Feingold (2000) made a critical observation that IBD-sharing statistics and correlation-based statistics are independent under both the null and alternative hypotheses, and so may be combined to give a more powerful test of linkage. The original proof of independence offered by Forrest and Feingold (2000) is overly complex and contains errors, but the result is correct. An alternative proof is included in the appendix A of this dissertation. On the basis of the independence theory, we can categorize various statistics into three groups based on the linkage information they use. The three groups are IBD-sharing statistics, correlation-based statistics, and combination statistics that combine the correlation and marginal IBD-sharing information, as previously defined in section 1.2.4. This important distinction among QTL mapping statistics is particularly essential for understanding which methods are appropriate for what kinds of samples.

One way to look at the relationship among the IBD-sharing statistics, the correlation-based statistics, and the combination statistics is to note that the computational formulas for the

statistics can be simply partitioned. Let π_i be the estimated IBD sharing for pair i, $\bar{\pi}$ the average IBD-sharing over all the pairs, and let A_i be some function of the sib's trait values (e.g. the squared trait difference for Haseman-Elston regression). The IBD-sharing statistic for a set of pairs is generally $\bar{\pi} - 1/2$, standardized by some appropriate variance estimate. The correlation-based statistics are generally of the form $\sum A_i(\pi_i - \bar{\pi})$, again standardized by some appropriate variance estimate, which is independent of $\bar{\pi} - 1/2$. Most of the combination statistics, by contrast, are of the form $\sum A_i(\pi_i - 1/2)$ (again, standardized). But $\sum A_i(\pi_i - 1/2)$ can be written as $\sum A_i(\pi_i - \bar{\pi}) + (\bar{\pi} - 1/2)\sum A_i$, showing that the combination statistics are a weighted sum of the correlation-based statistics and the IBD-sharing statistic. This decomposition is discussed in more detail in chapter 6.

2.2. DESCRIPTION AND THEORETICAL COMPARISON OF STATISTICS

In this section I describe in detail the test statistics considered in my dissertation. At the end of this section, I summarize the most important information of these statistics in tables 2 through 4, which are organized by the three categories introduced in section 2.1.

2.2.1. IBD sharing statistics

Sibling pairs that are ascertained based on a multiple-proband criterion are usually discordant and concordant pairs. They have very dissimilar or similar trait values and possess a property that makes them critically different from more typical samples – they have a distorted IBD sharing distribution at markers that are linked to the trait. A population sample of sibling pairs is expected to share half of their alleles IBD at any locus, regardless of whether that locus is linked to the trait

being studied. But in the case of discordant pairs, there should be substantially lower than expected IBD sharing at the marker linked to the trait; similarly, there should be higher than expected IBD sharing among the concordant pairs. So one tests for linkage by estimating the average number of alleles shared identical by descent (IBD) between the pairs at a marker and comparing it to the null hypothesis expectation. However, the IBD-sharing statistics are limited in that they only look at the IBD sharing, ignoring the actual trait values. Consequently, the pairs must be very extreme so that the IBD sharing distribution is distorted enough to achieve high power to detect linkage.

Risch and Zhang IBD-sharing statistic for EDSP (IBD1). Let π_i be the estimated mean IBD sharing for sibling pair *i*. The classical linkage test using EDSPs (Risch and Zhang 1995) uses the statistic

$$\frac{\overline{\pi} - 1/2}{\sqrt{\frac{1}{8n}}}.$$

The variance in the denominator is a theoretical value that assumes IBD information for each pair is observed perfectly (i.e., that the marker is infinitely polymorphic). This results in a conservative test when this statistic is applied to real data in which IBD sharing is estimated from marker data.

Robust Risch and Zhang IBD-sharing statistic (IBD2). Instead of the denominator used above, the IBD-sharing statistic can also be normalized using an empirical variance, which yields the statistic

$$\frac{\overline{\pi} - 1/2}{\sqrt{\sum (\pi_i - 1/2)^2 / n^2}}.$$

The test based on this statistic should have correct type I error even if the IBD information is not perfectly observed. One could also consider replacing the factor of 1/2 in the denominator with $\overline{\pi}$, which (named as *IBD3*) should result in a very slightly elevated type I error and power.

Gu et al. (1996) IBD-sharing statistic for EDAC. The "EDAC" test proposed by Gu et al. can be formulated as

$$\frac{\overline{\pi}_{CS} - \overline{\pi}_{DS}}{\sqrt{\frac{1}{8}\left[\frac{n_{CS} + n_{DS}}{n_{CS} \cdot n_{DS}}\right]}}$$

where n_{CS} is the total number of concordant pairs, n_{DS} is the number of discordant pairs, $\overline{\pi}_{CS}$, $\overline{\pi}_{DS}$ are the average estimated IBD sharing among concordant pairs and discordant pairs respectively, and $1/8$ is the variance among all the sib pairs assuming IBD information for each pair is observed perfectly.

Gu-empirical.variance for EDAC. We replaced the theoretical variance in the denominator of Gu et al. IBD-sharing statistic with an empirical variance estimate, designated as "Gu-empirical.variance" hereafter.

$$\frac{\overline{\pi}_{CS} - \overline{\pi}_{DS}}{\sqrt{\hat{\sigma}^2_{pooled}\left[\frac{n_{CS} + n_{DS}}{n_{CS} \cdot n_{DS}}\right]}}$$

where $\hat{\sigma}^2_{pooled}$ is the pooled empirical variance among all the sib pairs (assuming homoscedasticity under the null hypothesis). Similar to the above discussion for IBD2, Gu-empirical.variance should have correct type I error even when the marker is not completely informative.

2.2.2. Correlation-based statistics

As recently as a few years ago, the primary two statistical methods for QTL linkage in humans were both correlation-based: Haseman-Elston regression (Haseman and Elston 1972) and maximum likelihood variance components analysis (e.g. Amos 1994, Almasy and Blangero 1998). The Haseman-Elston method was derived under the assumption of a population sample of sib pairs with normally-distributed trait values, but the regression framework makes it quite robust to selected sampling and to non-Gaussian trait distributions. In contrast, variance components analysis is much more powerful than Haseman-Elston regression when trait distributions are approximately Gaussian, but is not very robust to selected sampling and deviations from distributional assumptions. (See Feingold 2001 and Feingold 2002 for more discussion). Hence, we do not include the variance components method in our comparisons as it is not appropriate for most selected samples (Allison et al.1999, Forrest and Feingold 2000).

Recently, a great deal of work has aimed to improve the power of Haseman-Elston regression method. New regression-based methods have been developed by Drigalenko (1998), Elston et al. (2000), Xu et al. (2000), Forrest (2001), Visscher and Hopper (2001), Sham and Purcell (2001), and Sham et al. (2002). All of these are based on linear regression, but use different functions of the trait values as the dependent variable. Several of these do in fact equal the power of variance components for population samples from "nice" trait models. While many of these methods should in theory be robust to selected sampling, most of the original papers did not specifically study selected samples.

Haseman-Elston (ORIGINAL.HE). Let x_{i1} and x_{i2} be the trait values for pair i, $Y_{iD} = (x_{i1}-x_{i2})^2$ be the squared trait difference for sibling pair i. The method of Haseman and Elston (1972) simply

regresses Y_{iD} on π_i and estimates the slope, $-\beta_D$. A positive estimate for β_D (a negative estimate for the slope) suggests that the trait is linked to the locus marker.

Trait-sum regression (TRAIT.SUM). Let μ be the population value of the trait mean, $Y_{iS} = [(x_{i1} - \mu) + (x_{i2} - \mu)]^2$ be the mean-corrected squared trait sum. We included the one-sided t-test of the slope β_S from the regression of Y_{iS} on π_i .

Trait-product regression (TRAIT.PRODUCT). Under population sampling, β_D and β_S are estimates of the same parameter (Drigalenko 1998). This slope parameter should be zero under the null hypothesis of no linkage, and should be positive (as we have defined the sign) under the alternative hypothesis. Drigalenko (1998) suggested averaging the two slope estimates, or , equivalently, doing a single regression with the mean-corrected trait product, $[(x_{i1} - \mu))(x_{i2} - \mu))]$, as the dependent variable. This method was futher developed by Elston et al.(2000). We consider the one-sided t-test based on the trait product regression.

Forrest's method (FORREST). Forrest (2001) suggested a test based on the weighted average

$$\hat{\beta} = \frac{\sigma_D^2}{\sigma_D^2 + \sigma_S^2}\hat{\beta}_S + \frac{\sigma_S^2}{\sigma_D^2 + \sigma_S^2}\hat{\beta}_D$$

where σ_D^2 and σ_S^2 are the variances of $\hat{\beta}_D$ and $\hat{\beta}_S$. These weights are optimal assuming that the covariance of $(\hat{\beta}_D, \hat{\beta}_S)$ is zero, which is true for a population sample from a normal distribution, but not necessarily otherwise (Feingold 2002). Forrest's method estimates all the parameters simultaneously using iterative least squares.

Visscher and Hopper's method (V&H). Visscher and Hopper (2001) proposed a test based on the same weighted slope estimate as Forrest (2001), but with the two variances estimated separately by performing the two regressions separately.

Xu et al.'s method (XU). Xu et al. (2000) proposed a method very similar to that of Forrest (2001) and Visscher and Hopper (2001), but their weights allow for a non-zero covariance between $\hat{\beta}_D$ and $\hat{\beta}_S$, i.e.

$$\hat{\beta} = \frac{\sigma_S^2 - \sigma_{DS}^2}{\sigma_D^2 + \sigma_S^2 - 2\sigma_{DS}^2}\hat{\beta}_D + \frac{\sigma_D^2 - \sigma_{DS}^2}{\sigma_D^2 + \sigma_S^2 - 2\sigma_{DS}^2}\hat{\beta}_S$$

This should perform much better than FORREST and V&H for non-Gaussian trait distributions and/or selected samples.

Sham and Purcell's method (HE-COM-correlation). The variances σ_D^2 and σ_S^2 can actually be calculated analytically as functions of the sibling trait correlation, *r*, under traditional QTL models. Sham and Purcell (2001) proposed taking advantage of this fact, rather than estimating the variances from data as in FORREST, V&H, and XU. The primary method outlined in Sham and Purcell (2001) regresses the dependent variable

$$\frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2}$$

on π_i, where the trait values x_{i1} and x_{i2} are standardized to mean zero and variance one before calculating Y_{iS} and Y_{iD}. Note that this statistic is called "S&P1" in the terminology of T.Cuenco et al. (2003) and Szatkiewicz et al. (2003).

2.2.3. Combination statistics

As pointed out by Forrest and Feingold (2000), the information in correlation-based statistics and IBD-sharing statistics is orthogonal, and statistics that combine both types of information can be very powerful for multiple-proband samples of various types. Besides Forest and Feingold's composite statistic (Forrest and Feingold 2000), there are also several statistics in the literature that combine information from both the marginal IBD sharing distribution and the correlation (Sham et al. 2000, Sham and Purcell 2001, Tang and Siegmund 2001, Wang and Huang 2002a, Putter and Sandkuijl 2002, Sham et al. 2002).

An important class of combination statistics is score statistics, which were originally proposed by Tang and Siegmund (2001), Wang and Huang (2002a), and Putter et al. (2002). Score statistics are based on the derivative of the usual variance components likelihood, but are more computationally convenient and can be constructed to be robust as well. Instead of considering precisely the statistics in the papers, we took the Tang and Siegmund (2001) statistic as our starting point and studied four variations on possible ways to make it robust (or not). Our first three score statistics below are constructed in such a way as to be combination statistics, while the last is constructed as a correlation-based statistic.

Asymptotic score statistic (SCORE1). Tang and Siegmund (2001) derived a score statistic of the form

$$\frac{\sum_i A_i(\pi_i - 0.5)}{\sqrt{2n\left[\frac{1+r^2}{(1-r^2)^2}\right]}},$$

where A_i is $\frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2} + \frac{4r}{1-r^2}$ for Y_{iS} and Y_{iD} are based on standardized trait values. The denominator of this statistic is based on asymptotic likelihood theory, so this version of the score statistic should *not* be robust to selected sampling or non-normality.

Score statistic with partially empirical variance (SCORE2). Tang and Siegmund (2001) proposed making their statistic robust by using the empirical variance of A_i in the denominator, i.e.

$$\frac{\sum_i A_i(\pi_i - 0.5)}{\frac{1}{2\sqrt{2}}\sqrt{\sum_i A_i^2}}$$

The factor of $1/2\sqrt{2}$ is the standard deviation of π assuming a perfectly informative marker. Thus, this version of the statistic should be robust to selected sampling, but should yield a conservative test when there is imperfect IBD information.

Score statistic with fully empirical variance (SCORE3). We proposed that the best version of the score statistic should have the same form as SCORE2, but with the empirical variance of π in place of the factor of $1/2\sqrt{2}$:

$$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - 1/2)^2\right]}}.$$

This version should have correct type I error even with imperfect IBD information. A slightly different alternative would be to replace the 1/2 in the denominator of SCORE3 with $\bar{\pi}$, which (named as *SCORE5*) would give very slightly higher type I error and slightly higher power than SCORE3 as we defined it.

Score statistic with empirical mean and variance (SCORE4). Both Wang and Huang (2002a) and Putter et al. (2002) proposed using $\bar{\pi}$ in place of 1/2 in both the numerator and denominator of the score statistic. Using $\bar{\pi}$ in the numerator creates a correlation-based statistic rather than a combination statistic. Applied to our parameterization of the score statistic, that yields the expression

$$\frac{\sum A_i(\pi_i - \bar{\pi})}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum (\pi_i - \bar{\pi})^2\right]}}.$$

We emphasize that SCORE4 is not identical to either Wang and Huang or Putter et al.'s statistics. Our SCORE4 should behave very similarly to SCORE3 in many cases, although in some situations it could have incorrect type I error because correlations between $\bar{\pi}$ and the π_i's are not accounted for in the denominator. That is, the denominator of SCORE4 is not actually the correct standard deviation of the numerator - there are missing covariance terms. It would also be possible to consider a score statistic that incorporates the covariance terms, but we did not include such a statistic in our study.

Yet another important combination statistic was proposed by Sham and Purcell (2001), which is a minor but important variation on HE-COM-correlation derived in the same paper.

Sham and Purcell's robust method (HE-COM-combination). Sham and Purcell (2001) also suggested a variant of their correlation-based method, regressing A_i on π_i - 1/2, with the intercept fixed at zero. This t-statistic for the test of the regression slope is

$$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left\{\left[\sum A_i^2\right]\left[\sum (\pi_i - 1/2)^2\right] - \left[\sum A_i(\pi_i - 1/2)\right]^2\right\}}}.$$

Note that the HE-COM-combination statistic is very similar to SCORE3, but with a cross-product term subtracted from the denominator of SCORE3. That should yield a statistic with slightly higher type I error and power than SCORE3. This statistic is called "S&P2" by T.Cuenco et al. (2003) and Szatkiewicz et al. (2003).

The Forrest and Feingold (2000) "composite" statistic was originally proposed for discordant sib pairs and is simply a weighted sum of an IBD-sharing statistic and a correlation-based statistic. We have developed composite statistics for EDAC pairs and for concordant/affected sibling pairs.

Composite statistic for discordant pairs (COMPOSITE1). Forrest and Feingold (2000) proposed to test for linkage using a weighted average of ORIGINAL.HE and IBD1:

$$w_{HE}\frac{-\hat{\beta}_D}{\hat{\sigma}_D}+w_{IBD}\frac{\overline{\pi}-1/2}{\sqrt{1/(8n)}},$$

where w_{HE} and w_{IBD} are arbitrarily-chosen weights. Based on limited calculations they recommended analyzing MDSPs using equal weights and EDSPs using a higher weight on the IBD-sharing statistic. Any of the correlation-based statistics can be used in place of ORIGINAL.HE in the composite. We used ORIGINAL.HE because it is the most powerful among all correlation-based statistics to date for discordant sib pairs. Note that the IBD-sharing component of the composite statistic is normalized using the theoretical variance, so it is expected to be conservative when there is imperfect IBD-sharing information.

Empirical composite statistic for discordant pairs (COMPOSITE2). The composite statistic for discordant pairs can also be formed as the average of ORIGINAL.HE and IBD2 (instead of

IBD1). This version should have correct type I error even when there is not perfect IBD information.

Based on the theory of combination statistics derived in section 2.1., we proposed a new combination statistic that is better than all existing methods for discordant sibling pairs – the robust discordant pair (RDP) statistic. Chapter 6 is devoted to describing this statistic in detail and comparing it to other statistics.

Robust discordant pair (RDP) statistic. This statistic is exactly the same as SCORE3, but with $(x_{i1} - x_{i2})^2$ substituted for A_i, i.e.

$$\frac{\sum (x_{i1} - x_{i2})^2 (\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left[\sum (x_{i1} - x_{i2})^4\right]\left[\sum (\pi_i - 1/2)^2\right]}}.$$

Consequently, RDP is parameter free and it should have the best features of both SCORE3 and COMPOSITE2

Composite statistic for concordant pairs. The composite statistic for concordant pairs can be formed as a weighted average of HE-COM-correlation and IBD2. Note that HE-COM-correlation is the most powerful among all correlation-based statistics to date for concordant sibling pairs. Chapter 8 has more details about this statistic.

For the discordant pairs and concordant pairs combined, we developed two composite statistics. Chapter 7 gives more details about these two statistics.

RDP composite statistics for EDAC pairs. We propose the following as the RDP-composite statistic for EDAC pairs to extract most significant linkage information from discordants and the concordants:

$$w_1 RDP_{discordant} + w_2 IBD_{concordant} \ ,$$

where $RDP_{discordant}$ is the "robust discordant pairs" statistic (Szatkiewicz and Feingold 2004) and is most powerful statistic for discordant pairs; $IBD_{concordant}$ is the robust mean IBD-sharing statistics for the concordant pairs; the weights, $w_{i,}$ are arbitrarily-chosen to combine the two components statistics. Good weights reflect the relative strength of each component and also adjust the signs so that the absolute values of each component are summed.

3-part composite statistic for EDAC pairs. Note that we can further decompose the RDP statistic and construct a composite statistic to be a weighted sum of three components: the IBD-sharing statistic for the discordant pairs, the IBD-sharing statistic for the concordant pairs, and original Haseman-Elston regression performed on the discordant pairs only, i.e.

$$w_1 ORIGINAL.HE_{discordant} + w_2 IBD_{discordant} + w_3 IBD_{concordant} \ .$$

The component IBD-sharing statistics appearing in both formulas above should be computed using empirical denominator as in statistic IBD2. Based on extensive testing, we concluded that a component measuring correlation for the concordant pairs should not be included in either of the above composite statistics because it adds more noise than signal in most cases.

For each of the composite statistics, it is also possible to use a chi-squared type statistic that essentially estimates the optimal weights from particular datasets. Appendix B provides an outline of construction of such kinds of composite statistics.

Two other combination statistics were not included in our simulation study due to computational limitations. One is the ascertainment-corrected variance components statistic proposed by Sham et al. (2000), which conditions on trait values. This statistic should perform very similarly to SCORE3 and HE-COM-combination. The other statistic that we did not include is the regression-based statistic proposed by Sham et al. (2002). This statistic was developed for extended pedigrees, but for sibling pairs it takes exactly the same form as SCORE2 and SCORE3, except that the variance of π is estimated differently.

2.2.4. Dependence of statistics on trait parameters

Many of the statistics we evaluated depend on estimates of trait parameters. The statistics TRAIT.SUM, TRAIT.PRODUCT, XU, V&H, FORREST, HE-COM-correlation, HE-COM-combination, SCORE1, SCORE2, SCORE3, and SCORE4 all use an estimate of the trait mean, μ. The S&P statistics and the SCORE statistics additionally use estimates of the trait variance, σ^2, and the sibling correlation, r. In general, theory suggests that these should be population values of the parameters, even when we are using selected samples. However, if one is using a selected sample, population parameter estimates may not be available. In that situation parameter values must be guessed or adopted from previous studies in other populations. Sensitivity to these parameter estimates can have an important effect on power. On the other hand, all IBD-sharing statistics, Haseman-Elston (1972), the RDP statistic, and all composite statistics except that for concordant pairs are parameter free and should be robust to model parameter misspecification.

Table 2 Summary of IBD-sharing statistics

Abbreviate name in dissertation	References	Formula	Assumptions	Comments
For discordant *or* concordant pairs				
IBD1	Risch and Zhang (1995)	$\frac{\overline{\pi} - 1/2}{\sqrt{1/8n}}$	Perfect IBD information	Conservative with estimated IBD
IBD2	Chapter 5, Chapter 8	$\frac{\overline{\pi} - 1/2}{\sqrt{\sum(\pi_i - 1/2)^2 / n^2}}$	-	Robust to imperfect IBD information
IBD3	Chapter 5	$\frac{\overline{\pi} - 1/2}{\sqrt{\sum(\pi_i - \overline{\pi})^2 / n^2}}$	-	Very slightly elevated type I error and power compared to IBD2
For EDAC samples				
Gu	Gu et. al. (1996)	$\frac{\overline{\pi}_{CS} - \overline{\pi}_{DS}}{\sqrt{\frac{1}{8}\left[\frac{n_{CS} + n_{DS}}{n_{CS} \cdot n_{DS}}\right]}}$	Perfect IBD information	Conservative with estimated IBD
Gu-empirical.variance	Chapter 7	$\frac{\overline{\pi}_{CS} - \overline{\pi}_{DS}}{\sqrt{\hat{\sigma}^2_{pooled}\left[\frac{n_{CS} + n_{DS}}{n_{CS} \cdot n_{DS}}\right]}}$	-	Robust to imperfect IBD information

Note: The numerators of the IBD-sharing statistics are all in the form of $\overline{\pi} - 1/2$

Table 3 Summary of correlation-basd statistics

Abbrev. name in dissertation	Reference	Definition	Comments		
			Y_D	Y_S	Weights based on
REGRESSION METHODS					
ORIGINAL.HE	Haseman & Elston (1972)	Conduct a single regression on $\pi_{i.}$ with the squared trait differences for each sibling pair (Y_{iD}) as dependent variable and estimate the slope ($-\beta_D$).	+	-	Binary
TRAIT.SUM	Chapter 4	Conduct a single regression on $\pi_{i.}$ with the mean-corrected squared trait sums for each sibling pair (Y_{iS}) as dependent variable and estimate the slope ($-\beta_S$).	-	+	Binary
TRAIT.PRODUCT	Drigalenko (1998), Elston et al. (2000)	Conduct a single regression on $\pi_{i.}$ with the mean-corrected trait product for each sibling pair (Y_{iP}), as dependent variable and estimate the slope (β_P).	+	+	Binary (equal weights)
V&H	Visscher & Hopper (2001)	Conduct two separate regressions on π_i once for Y_{iD}, once for Y_{iS} , form a weighted average of β_D and β_S with weights based on variance of β_D and β_S , where variances are estimated separately.	+	+	Empirical variance of β_D and β_S.
FORREST	Forrest (2001)	Form the same weighted slope estimate as Visscher and Hopper (2001), but with all the parameters are estimated simultaneously using iterative least squares.	+	+	Empirical variance of β_D and β_S.
XU	Xu et. al. (2000)	Conduct two separate regressions on π_i, form a weighted average slope based on both the variances of β_D and β_S and a non-zero covariance between them.	+	+	Empirical variance and covariance of β_D and β_S.
HE-COM-correlation	Sham & Purcell (2001)	Compute Y_{iD} and Y_{iS}, based on standardized trait values, form a weighted average of Y_{iD} and Y_{iS} with weights based on sibling trait correlation coefficient (r), conduct single regressions on π_i.	+	+	Population sibling correlation coefficient (r)
NON-REGRESSION METHODS					
SCORE4	Chapters 4, 5	Score statistic with pi-bar on both numerator and denominator.	+	+	Data-dependent automatic weights.
VC	e.g. Amos (1994)	Variance components	Not robust to any selected sample		

Note: The numerators of all the correlation-based statistics except VC are in the form of $\sum A_i(\pi_i - \bar{\pi})$, where A_i is some function of trait values with or without trait parameters involved.

Table 4 Summary of combination statistics

Abbrev. name in dissertation	References	Formula	Assumptions/requirements			
			Asymptotics	Normality	Parameters	External weights
SCORE STATISTICS*						
SCORE1	Tang and Siegmund (2001)	$\frac{\sum_i A_i(\pi_i - 1/2)}{\sqrt{2n[(1+r^2)/(1-r^2)]^2}}$	Asymptotic likelihood, perfect IBD	+	+	-
SCORE2	Tang and Siegmund (2001)	$\frac{\sum_i A_i(\pi_i - 1/2)}{\frac{1}{2\sqrt{2}}\sqrt{\sum_i A_i^2}}$	Perfect IBD	+	+	-
SCORE3	Chapters 4,5	$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - 1/2)^2\right]}}$	-	+	+	-
SCORE5	Chapters 4,5	$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - \bar{\pi})^2\right]}}$	-	+	+	-
RDP	Chapter 6	$\frac{\sum(x_{i1} - x_{i2})^2(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left[\sum(x_{i1} - x_{i2})^4\right]\left[\sum(\pi_i - 1/2)^2\right]}}$	-	+	-	-
COMPOSITE STATISTICS						
For discordant *or* concordant pairs						
COMPOSITE1	Forest & Feingold (2000)	$w_1 original.HE + w_2 IBD1$	Perfect IBD	-	-	+
COMPOSITE2	Chapter 5	$w_1 original.HE + w_2 IBD2$	-	-	-	+
Composite for concordant	Chapter 8	$w_1 HE.COM.correlation + w_2 IBD2$	-	-	+	+
For EDAC pairs						
RDP-composite	Chapter 7	$w_1 RDP_{discordant} + w_2 IBD_{concordant}$	-	+	-	+
3part-composite	Chapter 7	$w_1 ORIGINAL.HE_{discordant} + w_2 IBD_{discordant} + w_3 IBD_{concordant}$	-	-	-	+
OTHER STATISTICS						
HE-COM-combination	Sham and Purcell (2001)	Regress A_i on π_i with intercept fixed at zero	Very similar to SCORE3			
VC-conditional	Sham et al. (2000)	Ascertainment-corrected variance components, conditioning on trait values.	Very similar to SCORE3			
"Reverse regression"	Sham et al. (2002)	Regress a function of trait values on IBD-sharing	Very similar to SCORE3			

Note:* The numerators of the score variants are all in the form of $\sum A_i(\pi_i - 1/2)$, where A_i is $\frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2} + \frac{4r}{1-r^2}$

3. SOFTWARE DEVELOPMENT

3.1. SOFTWARE AVAILABILITY

Various software implementations are available for the older QTL mapping methods such as Haseman-Elston regression and variance components. The S.A.G.E package (2002) incorporates a few of the newer correlation-based statistics, such as the Elston et al. (2000) regression statistic. A couple of other new regression methods are individually implemented by the authors who proposed them (e.g. Xu et al. 2000). The IBD-sharing statistics can be calculated by standard binary trait mapping software such as GeneHunter (Kruglyak et al. 1996), Allegro (Gudbjartsson et al. 1999), or Merlin (Abecasis et al. 2002). The method of Sham et al. (2002) is also implemented in Merlin. But most of the combination statistics and score statistics, as well as several of the correlation-based statistics, are not readily available in any software package.

3.2. SOFTWARE DEVELOPMENT

3.2.1. Overview

We wrote two utility programs for our simulation/comparison study, both using C/C++ language. Each program has been extensively tested for correctness.

The first program simulates QTL data for sibling pairs under various models and ascertainment schemes. It allows a variety of Gaussian or non-Gaussian trait models, any number

of marker alleles (though only one marker), arbitrary recombination fraction between the marker and the trait locus, and sixteen different ascertainment schemes. It also computes a number of quantities useful for study designs; such as genotyping/phenotyping sample sizes, total number of families screened to obtain the final sample, and average noncentrality parameter (NCP) per sib pair (Sham and Purcell 2001) that approximates the theoretical power of any simulated sample.

The second program implements all of the statistics we considered (including minor variants) for sibling pairs. It can be used to analyze a single dataset, or it can calculate power over any number of datasets simulated by the first program. The output displays the mean, variance and empirical power of each statistic. If EDAC sib pair samples are used, it also gives average ratio of discordants to concordants in the sample. By modifying an input parameter file, sensitivity studies can be easily conducted.

There are also several small R (http://cran.r-project.org) programs written for exploring optimal weights for different versions of our composite statistics.

All of the programs are attached as appendixes C through E. Section 3.3 provides stand-alone documentation for the main C/C++ programs and section 3.4 provides a documented example run.

3.2.2. Research contribution

Our programs are based on the existing C programs written by Dr. William Forrest and used by Forrest and Feingold (2000). Particularly, I wrote the simulation program and co-authored the statistical program with Dr. Karen T.Cuenco. And I also wrote all of the R code.

3.3. PROGRAM DOCUMENTATION

3.3.1. Simulation program “Newsimsib5.c”

Description: C++ program “Newsimsib5.c” simulates QTL data for sibling pairs under various mixture-of -normals (Gaussian) models and ascertainment schemes. It allows any number of marker alleles of a single marker, arbitrary recombination fraction between the marker and the trait locus, and sixteen different ascertainment schemes. It also computes a number of quantities useful for study designs, i.e. the required genotyping/phenotyping sample sizes, total number of families screened to obtain the final sample. If desired, it calculates the average noncentrality parameter (NCP) per sib pair (Sham and Purcell 2001) that approximates the theoretical power of any simulated sample. However, this feature (NCP calculation) is not well tested. For maximum accuracy, statistical power should be evaluated empirically using simulation.

Compiling: “Newsimsib5.c” requires the following three header files, “IBD.c”, “random.h”, and “simplex.h” to be included in the same working directory.

(1) "IBD.c" is a utility file which takes genotype information at a single marker for two parents and two children and returns the estimated IBD sharing of the two children. (2) "random.h" is a utility file which combines several random number generators, including those for the random uniform, multinomial, standard normal and other variables. (3) "simplex.h" has a minimization routine taken from "Numerical Recipes in C".

To compile “Newsimsib5.c”, type "`g++ Newsimsib5.c -o Newsimsib5.out`". Note that the name of the executable (`Newsimsib5.out`) may be changed as one whishes.

***Available ascertainment schemes*:** “Newsimsib5.c” has implemented sixteen ascertainment schemes based on threshold selection. For each scheme, selection percentiles can be arbitrarily

specified by users, e.g., top 10 (x) percent and bottom 10 (y) percent for selecting discordant pairs. Table 5 provides a complete list of the sixteen ascertainment schemes.

Empirical cutoff values: "Newsimsib5.c" generates an empirical distribution of trait values under any mixture-of-normals model and computes the actual cutoff values that go with the x^{th} and y^{th} percentiles. In "Newsimsib5.c", 100,000 points are generated to simulate that empirical distribution. Note that for fuzzy concordant, the empirical cutoffs are based on the distribution of trait-plus-noise and ascertain pairs for whom trait-plus-noise is over the cutoff for both people.

Program inputs: For the first time when you run simulation, you will be instructed to input all the necessary parameters at the unixs prompt. "Newsimsib5.c" will generate an output file called "simulation_parameters" for you to record all of the values of those parameters you have used. In any future simulation, you may choose to input a new set of simulation parameters or to take advantage of the existing file "simulation_parameters". If you want to use the old values stored in the file "simulation_paraemters", you must include this file in the same working directory as an input file. If you only make minor changes of those old parameters you previously used, you may modify the file "simulation_parameters" directly and then include it as input file. Please be sure not to change any format of the file!! The format of the output/input file "simulation_parameters" is described in "Program outputs".

Program outputs: The simulation program "Newsimsib5.c" generates three output files "simulation_parameters", "sib_simulate","noise_simulate" by default. "sib_simulate" is the file to be read in by the statistical program ("CalSib8.c").

The first output file is "simulation_parameters", which stores the parameters you have used for your simulation. This file contains the following information:

```
line 1:  QTL gene frequency
line 2:  Recombination fraction
line 3:  number of marker allele
line 4:  population means for DD/Dd/dd
line 5:  population standard deviations for DD/Dd/dd
line 6:  environmental correlation
line 7:  type of ascertainment
line 8:  yth for lower_tail &  xth for upper_tail
line 8-2: y2th for lower_tail2 & x2th for upper_tail2
line 9:  noise_mu, noise_standard_deviation
line 10:  empirical cutoffs low_tail &  upper_tail
line 10-2:  empirical cutoffs low_tail2 &  upper_tail2
line 11: the number of replicates

Note:
a) Multiple numbers within each line are separated by spaces.
b) Line 8-2 and line 10-2 will appear only for ascertainment 8,9,10.
```

The second output file is "sib_simulate", which contains 3 parts in the following order:

(1) Trait values of each sibling pair and their probabilities of sharing 0, 1, 2 alleles IBD.

(2) "-999" immediately following part 1 in a new line, which signals the end of part 1 of the output file. (3) Records of the simulation parameters that have been used to generate trait values in part 1; records of sample sizes; and records for NCP calculation if that is applied. To see the last two parts of output file, type "`tail -25l sib_simulate`".

The last output file "noise_simulate" is only useful if you choose the fuzzy ascertainment schemes. "noise_simulate" is very similar to "sib_simulate" in the three part structure. However, the following information is recorded in the part 1 of "noise_simulate": Columns 1-2 are the noise values for each pair; columns 3-4 are the QTL trait values for each pairs; columns 5-6 are the trait-plus-noise values for the pairs

Technical information: If you are interested in more technical information regarding primary variables and the subroutine library, please consult the embedded documentation at the beginning of program "Newsimsib5.c" (Appendix C.)

Table 5 Ascertainment schemes implemented in the simulation program

Short name of ascertainment scheme	Definition	# in the selection menu
Population sampling	A random sample of sibling pairs , no selection at all	0
Single proband sampling		
Single.proband-both.sibs (top tail)	Phenotype both sibs in each pairs screened. Select all sib pairs where at least one sib is above the *x*th percentile.	1
Single.proband-both.sibs (bottom tail)	Phenotype both sibs in each pairs screened. Select all sib pairs where at least one sib is below the *y*th percentile.	2
Single.proband-both.sibs (both tail)	Phenotype both sibs in each pairs screened. Select all sib pairs where at least one sib is below the *y*th percentile or above the *x*th percentile.	3
Single.proband-one.sib (top tail)	Phenotype only the first sib in each pairs screened. Select all sib pairs where that first sib is above the *x*th percentile.	13
Single.proband-one.sib (bottom tail)	Phenotype only the first sib in each pairs screened. Select all sib pairs where that first sib is below the *y*th percentile.	14
Single.proband-one.sib (both tail)	Phenotype only the first sib in each pairs screened. Select all sib pairs where that first sib is below the *y*th percentile or above the *x*th percentile.	15
Multiple proband sampling		
Discordant	Select all sib pairs where one sib is above the *x*th percentile and the other is below the *y*th percentile.	4
Concordant high	Select all sib pairs where both siblings are above the *x*th percentile.	5
Concordant low	Select all sib pairs where both siblings below the *y*th percentile.	6
Concordant high and low	Select all sib pairs where both siblings below the *y*th percentile or both are above the *x*th percentile.	7
EDAC-3.corners (top)	Combined discordant pairs and concordant high pairs.	8
EDAC-3.corners (bottom)	Combined discordant pairs and concordant low pairs.	9
EDAC-4.corners	Combined discordant pairs and concordant low and high pairs.	10
Fuzzy.concordant (top tail)	Select all the pairs for which the values of trait-plus-noise of both sibs are above the *x*th percentile	11
Fuzzy.concordant (low tail)	Select all the pairs for which the values of trait-plus-noise of both sibs are below the *y*th percentile	12

3.3.2. Simulation program “Newsimsib_nonnormal.c”

C++ program “Newsimsib_nonnormal.c” is a variant of “Newsismib5.c” and it simulates sibling pair data from non-normal models. Currently, this program first simulates any mixture-of-normals traits exactly the same way as Newsimsib5.c does; it then calculates the signed square ($x|x|$) of these original trait values to form substantially non-normal trait distributions. Therefore, when you input simulation parameters, i.e., means and variances of the three genotypes, you should input the values of these parameters for the original mixture of-normals models. If you search for the key words "nonnormal traits" in “Newsimsib_nonnormal.c”, you can find the two sections where this signed square transformation has been implemented. Note that “Newsimsib_nonnormal.c” is a simplified version of Newsimsib5.c. It does not implement the single.proband-one.sib strategies; and it does not compute genotyping/phenotyping sample sizes, or average_NCP per sib pair. All other features of “Newsimsib_nonnormal.c” are the same as those of the program “Newsimsib5.c”.

3.3.3. Statistical program “CalSib8.c”

Brief description: C++ program “CalSib8.c” implements 36 QTL mapping statistics for sibling pairs. Table 6 provides a complete list of these statistics and their references. Please consult chapter 2 and the listed references for more detailed description of each statistic. “CalSib8.c” can be used to analyze a single dataset, or it can calculate empirical power over any number of datasets simulated by our simulation programs (“Newsimsib5.c” or “Newsimsib_nonnormal.c”). Furthermore, sensitivity studies of parameter misspecification can be easily conducted by modifying the input file “population_file” or by entering parameter estimates directly.

Compiling: First of all, the program “CalSib8.c” *and* the three header files, “IBD.c”, “random.h”, and “simplex.h” should be located in the same working directory. Next, type

"`g++ CalSib8.c -o CalSib8.out`" to compile "CalSib8.c". Note that the name of the executable ("`CalSib8.out`") can be freely changed.

Program inputs: "CalSib8.c" requires three input files: "sib_simulate", "CalSib8_input_file", "population_file" to be included in the same working directory. The three files are:

(1) "sib_simulate" stores the trait values and IBD information of each sibling pair. The output file "sib_simulate" generated by our simulation program ("Newsimsib5.c" or "Newsimsib_nonnormal.c") is ready to use. If analyzing any out-sources dataset which is not generated by any of our two simulation programs, the data matrix should be formatted and named exactly the same as "sib_simulate".

(2) "CalSib8_input_file" contains the following three lines:

```
line 1: number of pairs in each dataset (num_fam)
line 2: number of datasets (N_sample)
line 3: choice code for how to input the trait parameters that are required
by some statistics
```

Note for the choice code in line 3:

`112` – choice "*p*", i.e., input from file "population_file", see the item below

`117` – choice "*u*", i.e., input by users from screen at unixs prompt as instructed

`115` – choice "*s*", i.e., calculate sample estimates of the trait parameters

(3) If the choice is "`112`", i.e. input from "population_file", we need to include "population_file" as an input file in the same directory as well. The file "population_file" contains the following three lines:

```
line 1: overall trait mean
line 2: overall trait variance
line 3:  overall sibling pair correlation
```

Program outputs: "CalSib8.c" produces three output files: "statistics", "powers", "user_parameters".

(1) The file "statistics" is a data matrix of 36 (total number of statistics) by `N_samples` (total number of datasets), where each column is an array for one of the 36 QTL mapping statistics. This file can be imported into any statistical package for future examination. These statistics are:

- For any of the regression methods: t-statistic of the regression slopes;
- For any of the score statistics and variants: Z-score;
-For any of the IBD-sharing statistics: Z-score;
-For any of the composite statistics: Z-score;
-For any of the maximized composite statistics: chi-squared variable;
-For the variance component method: negative of the twice of the log-likelihood.

(2) The file "powers" consists of four columns:

-Column 1: the list of all 36 statistics

-Column 2: the empirical power of each statistic evaluated over all of the datasets (the number of datasets is specified by variable "`N_samples`")

-Column 3: the mean of each statistic evaluated over all the datasets

-Column 4: the standard deviation of each statistic evaluated over all the datasets

(3) The output file "user_parameters" stores the contents of "CalSib8_input_file" for checking or reminder purpose.

Technical information: If you are interested in more technical information regarding primary variables and the subroutine library, please consult the embedded documentation at the beginning of program "CalSib8.c" (Appendix D.)

Table 6 A complete list of statistics implemented in the statistical program

Name in program (in dissertation)*	Reference	Brief description	Calculated in function
VC	e.g. Amos (1994)	Variance components	Vc()
ORIGINAL.HE	Haseman &Elston (1972)	Regress Y_{iD} (squared trait difference) on $\pi_{i.}$	regression()
TRAIT.SUM		Regress Y_{iS} (mean-corrected squared trait sum)on π_i	regression()
TRAIT.PRODUCT	Elston et.al. (2000)	Regress Y_{iP} (mean-corrected trait product) on $\pi_{i.}$	regression()
V&H	Visscher & Hopper (2001)	Conduct two separate regressions on π_i , form a weighted average of β_D and β_S with weights based on their variance	Xu()
FORREST	Forrest (2001)	As Visscher and Hopper (2001) except with all the parameters are estimated simultaneously using iterative least squares.	Forrest()
XU	Xu et. al. (2000)	Form a weighted average of β_D and β_S based on both their variances and covariance between them.	Xu()
S&P1 (HE-COM-correlation)	Sham & Purcell (2001)	Regress $\frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2}$ on π_i. with trait values are standardized first	regression()
S&P2 (HE-COM-combination)	Sham & Purcell (2001)	Regress $\frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2} + \frac{4r}{1-r^2}$ on π_i - 1/2, with the intercept fixed at zero.	robustShamreg()
IBD1_neg	Risch and Zhang (1995)	Mean IBD-sharing statistic with theoretical variance, for discordant pairs	main()
IBD1_pos		Mean IBD-sharing statistic with theoretical variance, for concordant pairs	main()
IBD2_neg	Chapter 5	Mean IBD-sharing statistic with empirical variance, for discordant pairs	main()
IBD2_pos	Chapter 8	Mean IBD-sharing statistic with empirical variance, for concordant pairs	main()
IBD3_neg	Chapter 5	Mean IBD-sharing statistics with fully empirical variance (pi-bar in the denominator), for discordant pairs.	main()
IBD3_pos	-	Mean IBD-sharing statistics with fully empirical variance (pi-bar in the denominator), for concordant pairs.	main()
Gu	Gu et al. (1996)	Mean IBD-sharing statistic with theoretical variance. For combined discordant and concordant (EDAC) pairs.	main()
Gu-empirical.variance	Chapter 7	Mean IBD-sharing statistic with empirical variance For combined discordant and concordant (EDAC) pairs.	main()

Table 6 (continued)

Name in program (in dissertation)*	Reference	Brief description	Calculated in function
SCORE1	Tang and Siegmund (2001)	Asymptotic score statistic	Tang()
SCORE2	Tang and Siegmund (2001)	Score statistic with partially empirical variance	Tang()
SCORE3	Chapter 4	Score statistic with fully empirical variance	Tang()
SCORE4	Chapter 4	Score statistic with empirical mean and variance (correlation-based score statistic)	Tang()
SCORE5	Chapter 4	SCORE3 with pi-bar in the denominator	Tang()
SCORE6 (RDP statistic)	Chapter 6	SCORE 3 replacing A_i with square trait difference	Tang()
COMPOSITE_DS1	Forrest and Feingold (2000)	Weighted average of IBD1 and ORIGINAL.HE, equal weights implemented.	main()
COMPOSITE_DS2	Chapter 5	Weighted average of IBD2 and ORIGINAL.HE, equal weights implemented.	main()
COMPOSITE_DS3	Chapter 5	Weighted average of IBD3 and ORIGINAL.HE, equal weights implemented.	main()
COMPOSITE_CS2	Chapter 8	Weighted average of IBD2 and S&P1, equal weights implemented.	main()
COMPOSITE_CS3	Chapter 8	Weighted average of IBD3 and S&P1, equal weights implemented.	main()
COMPOSITE_DAC11	-	Weighted average of IBD2 plus S&P1 on entire sample, equal weights implemented.	main()
COMPOSITE_DAC12	-	Weighted average of IBD2 minus S&P1 on entire sample, equal weights implemented.	main()
COMPOSITE_DAC2 (4 part composite)	-	Weighted (optimally) sum of 4 components including: IBD2 and ORIGINAL.HE on discordant pairs, IBD2 and S&P1 on concordant pairs.	main()
COMPOSITE_DAC3 (3part-composite)	Chapter 7	Weighted (optimally) sum of 3 components including: IBD2 and ORIGINAL.HE on discordant pairs, IBD2 on concordant pairs.	main()
COMPOSITE_DAC6 (RDP-composite)	Chapter 7	Weighted (optimally) sum of RDP (score6) statistic on discordant pairs and IBD2 on concordant pairs.	main()
COMPOSITE_DS4 (Maximized 2part)	Appendix	Maximized COMPOSITE_DS2, for discordant pairs, $\chi^2(2)$	main()
COMPOSITE_DAC4 (Maximized 3part)	Appendix	Maximized COMPOSITE_DAC3, for EDAC pairs, $\chi^2(3)$	main()
COMPOSITE_DAC5 (Maximized 4part)	Appendix	Maximized COMPOSITE_DAC2, for EDAC pairs, $\chi^2(4)$	main()

Note: * The names appeared in ( ) are the names used in this dissertation.

3.4. EXAMPLE RUN

3.4.1. Simulation using new parameters

Simulation with new parameters requires the following four steps:

Step 1) Include "Newsimsib5.c" (or "Newsimsib_nonnormal.c" for non-normal models) *and* the three header files "IBD.c", "random.h", "simplex.h" in the same working directory.

Step 2) Compile "Newsimsib5.c" and run the executable "Newsimsib5.out".

Step 3) At the unixs prompt type the following commands and input your parameters as instructed. In the example run followed, we simulate trait values under model 1 of table 7 (on page 72 of this dissertation) with recombination fraction theta set to 0 to evaluate power (set theta = 0.5 to evaluate type I error). And we simulate 10,000 EDAC pairs aiming for a study of 1000 datasets with 100 pairs in each dataset. Note that everything that you must type/enter at the unixs prompt is in bold.

```
(1) unixs2 $ g++ Newsimsib5.c -o Newsimsib5.out
(2) unixs2 $ ./Newsimsib5.out
Create new parameter file? (y/n)
y

Please input parameters for mixture-of-normals model
QTL gene frequency:
0.1
Recombination fraction:
0
Number of (single) marker allele:
8

Trait means (normal model) for genotypes DD/Dd/dd, D is for the QTL
mu_DD:
1
mu_Dd:
0
mu_dd:
-1

Trait standard deviations (normal model) for genotype DD/Dd/dd
sd_DD:
0.849
sd_Dd:
```

0.849
sd_dd:
0.849

within-family sibling shared environmental correlation:
0.25

Types of ascertainment:

0) non-ascertained population sample
1) single.proband-both.sibs (top tail)
   - Phenotype both sibs in each pair screened and select all sib pairs where at least one sib is above the xth percentile.
2) single.proband-both.sibs (bottom tail)
   - Phenotype both sibs in each pair screened and select all sib pairs where at least one sib is below the yth percentile.
3) single.proband-both.sibs (two-tailed)
   - Phenotype both sibs in each pair screened and select all sib pairs where at least one sib is below the yth percentile or above the xth percentile.
4) discordant pairs
   - Select all sib pairs where one sib is above the xth percentile and the other is below the yth percentile.
5) high concordant pairs
   - Select all sib pairs where both sibs are above the xth percentile.
6) low concordant pairs
   - Select all sib pairs where both sibs are below the yth percentile.
7) low + high concordant
8) EDAC-3.corners-top: discordant + high concordant
9) EDAC-3.corners-bottom: discordant + low concordant
10) EDAC-4.corners: discordant + high and low concordant
11) fuzzy concordant high
   - Select all sib pairs where the values of trait-plus-noise for both sibs are above the xth percentile.
12) fuzzy concordant low
   - Select all sib pairs where the values of trait-plus-noise for both sibs are below the yth percentile.
13) single.proband-one.sib (top tail)
   - Phenotype only the first sib in each pair screened and select all sib pairs where that 1st sib is above the xth percentile.
14) single.proband-one.sib (bottom tail)
   - Phenotype only the first sib in each pair screened and select all sib pairs where that 1st sib is below the yth percentile.
15) single.proband-one.sib (two-tailed)
   - Phenotype only the first sib in each pair screened and select all sib pairs where that 1st sib is below the yth percentile or above the xth percentile.

8

Please enter the criteria to select discordant pairs first:
Please enter the desired xth percentile of upper_tail for discordance
12

```
Please enter the desired yth percentile of lower_tail for discordance
12
y 12 x 12

Please enter the criteria to select concordant pairs:
Please enter the desired x2th percentile of upper_tail for concordance
4
y2 99 x2 4

Top tail and bottom tail for discordance
-1.9031144  0.3244430
0.12 quantile: -1.9031144; 0.88 quantile: 0.3244430

Concordant top tail
0.9338698
0.96 quantile: 0.9338698; 0.96 quantile: 0.9338698

 Please input the number of replicates desired:
100000

Do you want to compute NCP_per_sibpair based on Sham&Purcell(2001)? (y/n)
y

Please input population trait parameters for NCP calculation

population trait mean:
-0.8

population trait standard deviation:
0.95

population sibling trait correlation :
0.3

overall_heritability:
0.15


Done!

 0.488   << average_NCP_per_sibpair

-0.800  0.950  0.300  0.150  << input population parameters mean/std/corr/H^2

6145749     << total # of families (pairs) screened
100000     << total # of families (pairs) ascertained after examining
7608006     << total # of people phenotyped
200000     << total # of people genotyped

 62698   37302   <<the number of discordant/total_concordant pairs
     0   37302   <<the number of LowConcordant/HighConcordant pairs
```

Step 4) Congratulations! As indicated, the simulation is done and you should have the two output files in your directory: “sib_simulate” and “simulation_parameters”. To view the output file

"simulation_parameters", type "`cat simulation_parameters`". And to view tail of the output file "sib_simualte", type: "`tail -35l sib_simulate`".

```
(4) unixs2 $ cat simulation_parameters
  0.10
  0.00
     8
   1.00   0.00  -1.00
 0.849  0.849  0.849
  0.25
     8
    12      12
    99       4
 0.000  0.000
-1.9031144  0.3244430
3.6812687  0.9338698
100000

(5) unixs2 $ tail -35l sib_simulate

  1.4428   1.3564     0.00   0.00   1.00
 -2.0799   0.6089     0.00   1.00   0.00
  0.4470  -2.0011     0.00   0.00   1.00
  0.9194  -2.0656     1.00   0.00   0.00
  0.4174  -2.3558     0.00   0.00   1.00
-999

 0.100     <<QTL gene frequency
 0.000     <<Recombination fraction
     8     <<number of marker allele
  1.000  0.000 -1.000   <<population means for DD/Dd/dd
 0.849  0.849  0.849    <<population standard deviations for DD/Dd/dd
 0.250     << shared environmental correlation
 0.000    0.000   << noise parameters : mean, standard deviation
     8     <<ascertainment scheme
     12      12   <<yth and xth for selection
-1.9031144  0.3244430    <<low_tail & upper_tail

Additional parameters for the combined discordant and concordant samples
    99       4   <<y2th and x2th for concordance selection
3.6812687  0.9338698    <<low_tail & upper_tail
 62698   37302   <<the number of discordant/total_concordant pairs
     0   37302   <<the number of LowConcordant/HighConcordant pairs

  100000   <<the number of replicates
 6145749   <<total # of families (pairs) screened
  100000   <<total # of families (pairs) ascertained after examining
 7608006   <<total # of people phenotyped
  200000   <<total # of people genotyped

 0.488   << average_NCP_per_sibpair
-0.800  0.950  0.300  0.150  <<input population parameters mean/std/corr/H^2
(6) unixs2 $
```

3.4.2. Simulation using input file "simulation_parameters"

Simulation with input file "simulation_parameters" requires the following four steps:

Step 1) Include "Newsimsib5.c" (or "Newsimsib_nonnormal.c" for non-normal models) *and* the three header files "IBD.c", "random.h", "simplex.h" in the same working directory.

Step 2) Include the input file "simulation_parameters" in the same working directory.

You may use any text editor to modify "simulation_parameters", but please make sure not to change any format.

Step 3) Compile "Newsimsib5.c" and run the executable "Newsimsib5.out". At the unixs prompt type the following commands and type "n" to use the input file. In the example run followed, everything you must type in is in bold. Note that in this example, we choose not to compute the NCP per sibling pair.

```
(1) unixs2 $ g++ Newsimsib5.c -o Newsimsib5.out
(2) unixs2 $ ./Newsimsib5.out
Create new parameter file? (y/n)
n
reading exiting file
allele prequency:0.100000
Recombination fraction:  0.00
Number of marker allele:     8
means:  1.00   0.00  -1.00
sd: 0.849  0.849  0.849
environmental correlation:  0.25
type of ascertainment is :     8

yth percentile of lower_tail & xth percentile of upper_tail : 12     12
above criteria are for discordance; now the criteria for concordance:
y2th percentile of lower_tail2 & x2th of upper_tail2 :     99      4

noise_mu:  0.00, noise_sd:  0.00

low_high_tails:-1.9031144  0.3244430
above criteria are for discordance; now the criteria for concordance:
low_high_tails:3.6812687  0.9338698

The number of replicates desired:100000

Do you want to compute NCP_per_sibpair based on Sham&Purcell(2001)? (y/n)
n
```

```
Done!

6152694     << total # of families (pairs) screened
100000     << total # of families (pairs) ascertained after examining
7613999     << total # of people phenotyped
200000     << total # of people genotyped
 63018   36982   <<the number of discordant/total_concordant pairs
     0   36982   <<the number of LowConcordant/HighConcordant pairs
(3) unixs2 $
```

Step 4) Congratulations! As indicated, the simulation is done and you should have two output files in your directory: "sib_simulate" and "simulation_parameters". To view the output file "simulation_parameters", type "`cat simulation_parameters".` And to view tail of the output file "sib_simualte", type: "`tail -35l sib_simulate".`

3.4.3. Calculating statistics and evaluating empirical power

To calculate statistics and evaluate empirical power, we take the following steps:

Step 1) Include "CalSib8.c" *and* three header files "IBD.c", "random.h", "simplex.h" in the same working directory.

Step 2) Include "sib_simulate" in the same working directory. If it is the output file generated by our simulation programs ("Newsimsib5.c"or "Newsimsib_nonnormal.c"), it is ready to use. If using any out-sources data, please make sure that the data file is formatted and named exactly as the simulated data file "sib_simulate".

Step 3) Include appropriate input file "CalSib8_input_file" in the same working directory. "CalSib8_input_file" can be modified using any text editor. For example, if we want to do a study with 1000 datasets and 100 sib pairs in each dataset, and if we want to use the "population_file" (option code is `112`) to input known trait parameters that are required by some of the statistics, the input file should look like this:

```
(1) unixs2 $ cat CalSib8_input_file
100
1000
112
```

Step 4) Suppose, as we have done in step 3, we want to use input file "population_file" to read in known population parameters that are required by some of the statistics, we must include this "population_file" in the same working director. We can use any text editor to modify the parameters. For example, if the trait mean, variance and sibling correlation are -0.8, 0.9, 0.3, respectively, file "population_file" should look like this:

```
(2) unixs2 $ cat population_file
-0.8
0.9
0.30
```

Step 5) To compile "CalSib8.c" and run the executable "CalSib8.out", type the following commands at the unixs prompt.

```
(3) unixs2 $ g++ CalSib8.c –o CalSib8.out
(4) unixs2 $ ./CalSib8.out
p       112
Please check file <user_parameters> for reminder of the sample used
Please check file <powers> for power results
Total_n_DS, Total_n_CS: 63018   36982
```

Step 6) Congratulations! The calculation has been completed and you should have the three output files generated in your directory. Now you are ready to check the power (or type I error if using simulated data with theta=0.5) results.

Step 7) View the output file "powers" for power and a summary (mean and standard deviation) of each statistic. Note that in the following example, subroutine Vc() was commented out, i.e., variance components -2LL statistic was not computed.

```
(5) unixs2 $ cat powers

Statistic             Power    Average   Std
Variance_Components   0.000000 0.000000 0.000000
ORIGINAL.HE           0.798000 -3.211407 1.047911
TRAIT.SUM             0.794000 3.157969 1.043476
TRAIT.PRODUCT         0.843000 3.359107 1.059283
V&H                   0.585000 2.571473 1.100691
FORREST               0.924000 4.270845 1.309690
XU                    0.574000 2.529700 1.054870
S&P1                  0.839000 3.396638 1.063744
S&P2                  0.840000 3.373087 1.026250
SCORE1                0.997000 12.035325 3.556302
SCORE2                0.774000 2.955394 0.854769
SCORE3                0.823000 3.156652 0.871766
SCORE5                0.827000 3.177887 0.881513
SCORE6                0.523000 -2.316692 0.884904
SCORE4                0.780000 2.962818 0.854445
IBD1_pos              0.003000 -0.454245 0.951160
IBD1_neg              0.032000 -0.454245 0.951160
IBD2_pos              0.002000 -0.484958 1.011049
IBD2_neg              0.036000 -0.484958 1.011049
IBD3_pos              0.002000 -0.493082 1.030305
IBD3_neg              0.041000 -0.493082 1.030305
COMPOSITE_DS1         0.124000 -1.270555 0.924854
COMPOSITE_DS2         0.146000 -1.300223 0.979031
COMPOSITE_DS3         0.155000 -1.308072 0.996720
COMPOSITE_DS4(M2)     0.751000 12.440713 6.891367
COMPOSITE_CS2         0.410000 2.058869 1.097368
COMPOSITE_CS3         0.410000 2.053124 1.107830
COMPOSITE_DAC11       0.410000 2.058869 1.097368
COMPOSITE_DAC12       0.668000 -2.744703 0.974451
COMPOSITE_DAC2        0.825000 3.188749 0.890534
COMPOSITE_DAC3        0.812000 3.135463 0.887569
COMPOSITE_DAC6        0.839000 3.194923 0.893497
COMPOSITE_DAC4(M3)    0.735000 12.628379 5.659591
COMPOSITE_DAC5(M4)    0.704000 13.277673 5.728415
IBD1_EDAC             0.728000 2.833779 0.885208
IBD2_EDAC             0.781000 3.026786 0.908520

The number of families in each study: 100
The number of studies: 1000
```

Step 8) The output file “statistics” can put imported to any statistical package (e.g., R) for further examination. You can also view the output file “use_parameters”.

```
(5) unixs2 $ cat user_parameters
num_fam 100 N_samples 1000
choice of population_file, user_input or sample estimates: p
mean -0.8000 variance 0.9000 corr 0.3000
```

3.4.4. Conducting sensitivity study

Sensitivity studies of the effects of model parameter misspecification can be performed exactly the same way as what has been described in section 3.4.3. You can use pre-specified values of the trait parameters, or you can use sample estimates of the trait parameters.

Using pre-specified values: To use pre-specified values of the trait parameters, there are two ways to input them. (1) Modify the input file "population_file" as described above. (2) Input new parameters directly from screen, which can be done in two steps as below.

Step1) Modify "CalSib8_input_file" – change code 112 to 117 (for "*u*" user_input), as:

```
unixs2 $ cat CalSib8_input_file
100
1000
117
```

Step 2) Run the executable "CalSib8.out" and input parameters as instructed, for example:

```
(4) unixs2 $ ./CalSib8.out
u  117

Do you want to use population, user-input, or sample trait parameters?
(p/u/s)
u

Input parameters for model
trait mean:
-0.8
trait variance:
0.9
trait correlation:
0.5
```

Using sample estimates: To use sample estimate of the parameters, we do the two steps:

Step1) modify "CalSib8_input_file" – change code 112 to 115 (for "s" sample)

```
unixs2 $ cat CalSib8_input_file
100
1000
115
```

Step2) Run the executable "CalSib8.out".

4. RECENT ADVANCES IN HUMAN QUANTITATIVE-TRAIT-LOCUS MAPPING: COMPARISON OF METHODS FOR SELECTED SIBLING PAIRS.

This chapter has been published in *American Journal of Human Genetics* (T.Cuenco et al. 2003). I have obtained the copyright permission from The Chicago Press. The content of T.Cuenco et al. (2003) is used below without change but its format has been modified to fit this dissertation. Dr. Karen T.Cuenco co-authored the statistical program and performed all simulation studies for that paper. I wrote the simulation program, co-authored the statistical program, and also conducted preliminary simulation analyses for this project.

4.1. SUMMARY

In the last few years, there has been a great deal of new work on methods for mapping quantitative trait loci using sibling pairs and sibships. There are several new methods based on linear regression, and several more based on score statistics. In theory, most of the new methods should be relatively robust to violations of distributional assumptions and to selected sampling, but in practice there has been little evaluation of how the methods perform on selected samples. We survey most of the new regression-based statistics and score statistics, and propose a few minor variations on the score statistics. We use simulation to evaluate the type I error and power of all of the statistics, considering both population samples of sibling pairs and sibling pairs ascertained on the basis of at least one sibling having a trait value in the top 10% of the

distribution. Most, though not all, of the statistics have correct type I error for selected samples. The statistics proposed by Xu et al. (2000) and by Sham and Purcell (2001) are generally the most powerful, along with one of our score statistic variants. Even among the methods that are most powerful for "nice" data, some are more robust than others to non-Gaussian trait models and/or misspecified trait parameters.

4.2. INTRODUCTION

As recently as a few years ago there were only two primary statistical methods for quantitative trait locus (QTL) linkage analysis using sibships: Haseman-Elston regression (Haseman and Elston 1972) and maximum likelihood variance components analysis (e.g. Amos 1994, Almasy and Blangero 1998). The Haseman-Elston method was derived under the assumption of a population sample of pairs with normally-distributed trait values, but the regression framework makes it quite robust to selected sampling and to non-Gaussian trait distributions (see Feingold 2002 for a more extended discussion). Variance components analysis, on the other hand, has much higher power than Haseman-Elston regression under ideal conditions, but is not very robust to selected sampling and deviations from distributional assumptions (see Feingold 2001 for more discussion). Recently, there has been an explosion of new methods that aim to equal the power of variance components while retaining the robustness of Haseman-Elston regression. One set is regression-based statistics, essentially improvements on the original Haseman-Elston method. New regression-based methods have been developed by Drigalenko (1998), Elston et al. (2000), Xu et al. (2000), Forrest (2001), Visscher and Hopper (2001), Sham and Purcell (2001), and Sham et al. (2002). The other set of new methods is score statistics, based on the derivative

of the usual variance components likelihood. The three primary papers on these methods are Tang and Siegmund (2001), Putter et al. (2002), and Wang and Huang (2002a), with extensions in Tang and Siegmund (2002), Wang (2002), and Wang and Huang (2002b). The new score statistics are more computationally convenient than variance components, and they can be constructed in such a way as to be robust as well.

Theoretical reviews of most of the new regression-based and score-based statistics have appeared in Feingold (2001) and Feingold (2002). A number of other papers have compared limited subsets of these methods using theory or simulations, including Allison et al. (2000), Palmer et al. (2000), Goldstein et al. (2001), Visscher and Hopper (2001), Zhang et al. (2001), Ghosh and Reich (2002), and Zhang et al. (2002). Despite the fact that most of the new statistics should in theory be appropriate for selected samples, there has been very little actual testing on such samples. In this study we undertake a comprehensive simulation-based comparison of the new statistics. We limit ourselves to sibling pairs for simplicity, but many of the general results are applicable to larger sibships as well. We perform simulation studies using both population and selected samples and estimate the type I error and power of each statistic. We consider eleven different trait distributions, some of them substantially "non-Gaussian," and also consider robustness of the methods to misspecification of trait parameters. The selected samples considered in this chapter consist of sibling pairs ascertained on the basis of at least one sibling in the top 10% of the trait distribution. Note that there are important statistical differences between samples ascertained on the basis of a single individual and samples ascertained on the basis of more than one individual.

4.3. METHODS

4.3.1. Statistics considered

Here, we briefly define the twelve QTL mapping statistics that we consider in this chapter. More detailed description of the statistics can be found in the reviews by Feingold (2001) and Feingold (2002), as well as in the original papers cited below.

Some notation and definitions are common to all of the statistics. Let π_i be the estimated mean IBD sharing for sibling pair *i*; π_i takes values 0, 1/2, or 1 for a fully-informative pair, but can take intermediate values if multi-point estimates are used. Let $Y_{iD} = (x_{i1} - x_{i2})^2$ be the squared trait difference for sibling pair *i*. Analogously, define $Y_{iS} = [(x_{i1} - \mu) + (x_{i2} - \mu)]^2$, the mean-corrected squared trait sum. The regression of Y_{iD} on π_i produces a slope estimate. We define negative one times this slope estimate as $\hat{\beta}_D$, and let $\hat{\beta}_S$ be the slope estimate from a regression of Y_{iS} on π_i . Under population sampling, $\hat{\beta}_D$ and $\hat{\beta}_S$ are estimates of the same parameter (Drigalenko 1998). This slope parameter should be zero under the null hypothesis of no linkage, and should be positive (as we have defined the sign) under the alternative hypothesis. The first eight methods described below are all based on different methods of combining the information from these two regressions.

Original Haseman-Elston (ORIGINAL.HE). The original method of Hasman and Elston (1972) simply regresses Y_{iD} on π_i and estimates the slope which is equivalent to $-\hat{\beta}_D$. A negative estimate suggests that the trait is linked to the locus marker. A one-sided t-test is used to test for any significant departure from zero.

Trait-sum regression (TRAIT.SUM). For comparison to the other statistics, we include the one-sided t-test based on the regression of Y_{iS} on π_i, though we do not expect this statistic to be particularly powerful.

Trait-product regression (TRAIT.PRODUCT). Drigalenko (1998) suggested doing the two regressions described above and averaging the two slope estimates, or , equivalently, doing a single regression with the mean-corrected trait product, $[(X_{i1} - \mu)(X_{i2} - \mu)]$, as the dependent variable. This method was further developed by Elston et al. (2000). We consider the one-sided t-test based on the trait-product regression.

Forrest's method (FORREST). Rather than a simple average of the two regression slope estimates, it is more statistically desirable to use an average that is weighted by the variances of the estimates. Forrest (2001) suggested a test based on the weighted average

$$\hat{\beta} = \frac{\sigma_D^2}{\sigma_D^2 + \sigma_S^2}\hat{\beta}_S + \frac{\sigma_S^2}{\sigma_D^2 + \sigma_S^2}\hat{\beta}_D \ ,$$

where σ_S^2 and σ_D^2 are the variances of $\hat{\beta}_D$ and $\hat{\beta}_S$. These weights are optimal assuming that the covariance, σ_{DS}^2, of $(\hat{\beta}_D, \hat{\beta}_S)$ is zero, which is true for a population sample from a normal distribution, but not necessarily otherwise (Feingold 2002). Forrest's method estimates all the parameters simultaneously using iterative least squares.

Visscher and Hopper's method (V&H). Visscher and Hopper (2001) proposed a test based on the same weighted slope estimate as Forrest (2001), but with the variances, σ_S^2 and σ_D^2, estimated separately by performing the two regressions separately.

Xu et al.'s method (XU). Xu et al. (2000) proposed a method very similar to that of Forrest (2001) and Visscher and Hopper (2001), but their weighted average slope allows for a non-zero covariance between $\hat{\beta}_D$ and $\hat{\beta}_S$ using the formula

$$\hat{\beta} = \frac{\sigma_S^2 - \sigma_{DS}^2}{\sigma_D^2 + \sigma_S^2 - 2\sigma_{DS}^2}\hat{\beta}_D + \frac{\sigma_D^2 - \sigma_{DS}^2}{\sigma_D^2 + \sigma_S^2 - 2\sigma_{DS}^2}\hat{\beta}_S.$$

Xu et al. estimate the parameters by performing the two regressions separately, similarly to Visscher and Hopper. The covariance can be estimated by combining the residuals of the two regressions.

Sham and Purcell's method (HE-COM-correlation). The variances σ_D^2 and σ_S^2 can actually be calculated analytically as functions of the sibling trait correlation, *r*, under traditional QTL models. Sham and Purcell (2001) proposed taking advantage of this fact, rather than estimating the variances from data as in FORREST, V&H, and XU. The primary method outlined in Sham and Purcell (2001) regresses the dependent variable

$$\frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2}$$

on π_i , where the trait values x_{i1} and x_{i2} are standardized to have a variance of one before calculating Y_{iS} and Y_{iD}.

Sham and Purcell's robust method (HE-COM-combination). Sham and Purcell also suggested a variant of their method, regressing

$$A_i = \frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2} + \frac{4r}{1-r^2}$$

on π_i - 1/2, with the intercept fixed at zero. This variant should be more robust to selected sampling. The t-statistic for the test of the regression slope is

$$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left\{\left[\sum A_i^2\right]\left[\sum(\pi_i - 1/2)^2\right] - \left[\sum A_i(\pi_i - 1/2)\right]^2\right\}}}.$$

The final four methods we consider are score statistics based on the usual variance components likelihood. Score statistics were proposed by Tang and Siegmund (2001), Wang and Huang (2002a), and Putter et al. (2002). The score statistics proposed in those papers are very similar to each other, but have minor differences in how they parameterize the likelihood and how they alter the statistic to make it robust. Instead of considering precisely the statistics in the papers, we take the Tang and Siegmund (2001) statistic as our starting point and propose four variations on possible ways to make it robust (or not). This allows us to draw careful conclusions about what kind of "robustification" is most desirable.

Asymptotic score statistic (SCORE1). Tang and Siegmund (2001) derived a score statistic of the form

$$\frac{\sum_i A_i(\pi_i - 0.5)}{\sqrt{2n\left[\frac{1+r^2}{(1-r^2)^2}\right]}},$$

where A_i is the same function defined for HE-COM-combination above. The denominator of this statistic is based on asymptotic likelihood theory, so this version of the score statistic should *not* be robust to selected sampling or non-normality.

Score statistic with partially empirical variance (SCORE2). Tang and Siegmund (2001) proposed making their statistic robust by using the empirical variance of A_i in the denominator, i.e.

$$\frac{\sum_i A_i(\pi_i - 0.5)}{\frac{1}{2\sqrt{2}}\sqrt{\sum_i A_i^2}}.$$

The factor of $1/2\sqrt{2}$ is the standard deviation of π assuming a perfectly informative marker. Thus, this version of the statistic should be robust to selected sampling, but should yield a conservative test when there is imperfect IBD information.

Score statistic with fully empirical variance (SCORE3). We propose that the best version of the score statistic should have the same form as SCORE2, but with the empirical standard deviation of π in place of the factor of $1/2\sqrt{2}$:

$$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - 1/2)^2\right]}}.$$

This version should have correct type I error even with imperfect IBD information. A slightly different alternative would be to replace the 1/2 in the denominator of SCORE3 with $\bar{\pi}$, which (named as *SCORE5*) would give very slightly higher type I error and slightly higher power than SCORE3 as we have defined it. Note that the HE-COM-combination statistic described above is very similar to SCORE3, but with a cross-product term subtracted from the denominator. Again, that should yield a statistic with slightly higher type I error and power than SCORE3.

Score statistic with empirical mean and variance (SCORE4). Both Wang and Huang (2002a) and Putter et al. (2002) proposed using $\bar{\pi}$ in place of 1/2 in both the numerator and denominator of the score statistic. Applied to our parameterization of the score statistic, that yields the expression

$$\frac{\sum A_i(\pi_i - \bar{\pi})}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - \bar{\pi})^2\right]}}.$$

We emphasize that SCORE4 is not identical to either Wang and Huang or Putter et al.'s statistics. Our SCORE4 should behave very similarly to SCORE3 in many cases, although in some situations it could have incorrect type I error because correlations between $\bar{\pi}$ and the π_i's are not accounted for in the denominator. That is, the denominator of SCORE4 is not actually the correct standard deviation of the numerator - there are missing covariance terms. It would

also be possible to consider a score statistic that incorporates the covariance terms , but we did not include such a statistic in our study.

For the following discussion, it is useful to classify our twelve statistics into three groups. Group I uses simple binary weights of the two regression slopes (ORIGINAL.HE, TRAIT.SUM, TRAIT.PRODUCT). These methods are all expected to have sub-optimal power because of sub-optimal weighting. Group II uses empirical variances to weight the two slopes (FORREST, V&H, XU). Group III uses the sibling trait correlation to achieve weighted statistics without calculating empirical variance estimates (HE-COM-correlation, HE-COM-combination, SCORE1, SCORE2, SCORE3, SCORE4).

All of the statistics we consider, except for ORIGINAL.HE, use an estimate of the trait mean, μ. Group III statistics additionally use estimates of the trait variance, σ^2, and sibling correlation, r. Sensitivity to these estimates may have an important effect on power.

4.3.2. Simulations

We studied the type I error and power of each statistic under eleven trait models, which are described in table 7. All models are diallelic. Models one through nine are standard mixture-of-normals models; the trait value is equal to the genotype mean plus a normally-distributed "environmental" variance. There is an additional sibling correlation of .25 in each model to account for environmental and polygenic components. The means and variances were chosen to give each model a locus heritability of .2. Note the symmetry between certain pairs of models: 1 and 7; 2 and 9; 3 and 8; and 5 and 6. This symmetry means that type I error and power within each pair are identical for population samples, though not for selected samples. Models 1' and 2' were generated by simulating data under models 1 and 2, respectively, and then taking the signed

square ($x|x|$) of each trait value. This yields overall trait distributions that are somewhat skewed and have high kurtosis. Models 3 and 8 also have skewness and kurtosis in the same range as models 1' and 2'.

We simulated data for nuclear families with two children according to each of the models, and ascertained families by two different methods. The first ascertainment scheme was simply population sampling -- all families were used. The second scheme selected only those families in which at least one sibling fell in the top 10% of the trait distribution. We simulated datasets of 1500 families for the population sampling and 500 families for the selected sampling. Figure 2 shows examples of simulated bivariate trait distributions for both sampling schemes under several of the models. To study type I error, we used 10,000 datasets, and to study power we used 1,000 datasets. The nominal type I error rate was set at .01. Marker data was simulated using eight equifrequent alleles, with the marker at $\theta = 0$ for the power study and at $\theta = 1/2$ for the type I error study. We also did power simulations at $\theta = 0.05$ for models 1 and 2 only.

As discussed above, most of the statistics require that some trait parameters (mean μ, variance σ^2, sibling correlation r) be specified. In general, theory suggests that these should be population parameter values, even for selected samples. However, if one is using a selected sample, population parameter estimates may not be available. In that situation parameter values must be guessed or adopted from previous studies in other populations. We examined the robustness of the statistics to misspecification of parameters using models 1 and 1' only. We varied one parameter at a time while holding the other two parameters at the correct population values. Sibling correlation was set at 0.1 and 0.5, trait variance at values ranging from half the true value to twice the true value and trait mean at the true mean plus and minus one standard

deviation. We also did a limited number of studies with two parameters at a time misspecified. Finally, we checked the performance of the statistics using sample estimates of the parameters.

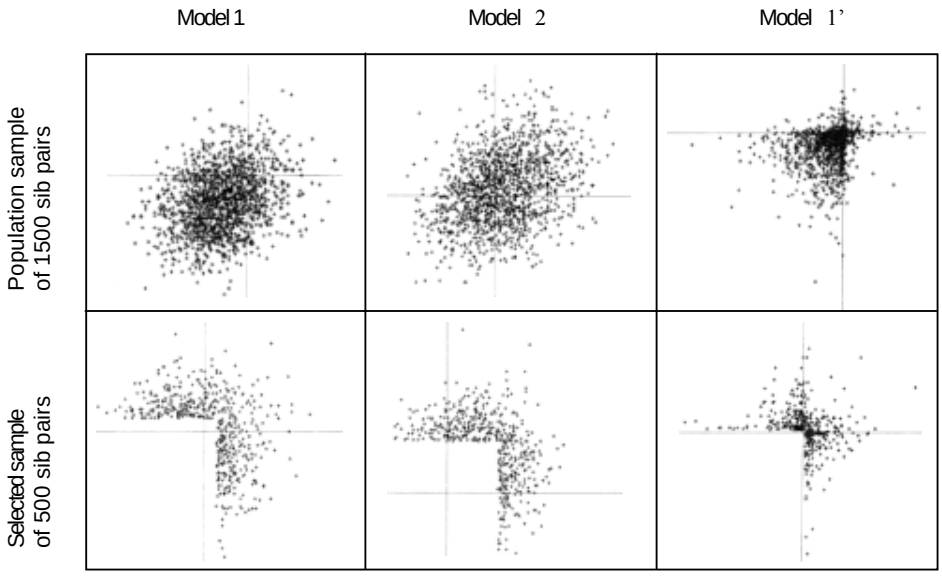

Figure 2 Scatter plots of population and selected samples from models 1, 2, and 1'.

4.4. RESULTS

4.4.1. Type I error

Table 8 shows the standard deviation and type I error of each statistic based on the 10,000 simulated datasets of population samples. Table 9 shows the same information for the selected samples. All statistics had mean zero for all models and all sampling schemes. We show only results for models 1, 2, 3, 1', and 2'. Results for models 4 – 9 were very qualitatively similar to those for models 1 – 3. All of the statistics in these tables were computed with the known *population* values of the parameters (trait mean μ, variance σ^2, and sibling correlation r). The confidence intervals for the estimated error rates in the tables are on the order of plus or minus 0.2% (i.e. an estimated error rate of 1.00% has a 95% confidence interval of approximately 0.80% to 1.20%). We note first that the type I error and standard deviation for SCORE1 and SCORE2 are incorrect for essentially all models and sampling schemes. SCORE2 is always conservative (low type I error) because of the perfect IBD assumption; SCORE1 is highly variable, presumably because the asymptotic normality assumption underlying it is inappropriate for many of these trait distributions. The V&H and FORREST statistics have incorrect type I error for the most non-Gaussian models (3, 1' and 2') under population sampling and for all models under selected sampling. This is due to the omission of the covariance term in the weighting. Finally, SCORE4 has low type I error for some models under selected sampling. This is because of the missing covariance in the denominator of the statistic; the size of the covariance term depends heavily on the value of the A_i's, and for some models and sampling schemes it can be quite large. As predicted, HE-COM-combination and SCORE3 are very similar, with HE-COM-combination having a slightly higher type I error rate for most models. We did limited

experiments (results not shown) with a version of SCORE3 that replaces the "1/2" in the denominator by $\overline{\pi}$ (see methods), and found that it has type I error rates just about identical to those of HE-COM-combination.

4.4.2. Power

Table 10 gives the power for all models for the population samples, and table 11 gives the power for the selected samples. Again, all of the statistics in these tables were computed with the known population values of the parameters. To make comparisons simpler, the statistics that did not have correct type I error are omitted from the power tables. The number of replicates for the power study was 1000, so the 95% confidence interval for a power estimate of 50% is approximately 47% to 53%. The general qualitative results are quite similar for the two sampling schemes. The group I statistics have lower power than the group II and group III statistics in almost all cases. This is attributable to the suboptimal weighting of the sum and difference regression slopes in the group I statistics. All of the group II and group III statistics have essentially identical power, with the exception that XU has noticeably higher power for models 1' and 2'. We did limited experiments (results not shown) with a version of SCORE3 that replaces the "1/2" in the denominator by $\overline{\pi}$ (see methods), and found, as predicted, that it has power rates just about identical to those of HE-COM-combination. We also did power simulations at $\theta = 0.05$ for models 1 and 2 only (results not shown); while the overall power is lower than at $\theta = 0$, the relative power of the different statistics is unchanged.

4.4.3. Sensitivity

All of the statistics except ORIGINAL.HE use estimates of the mean (μ) parameter. In addition, all of the group III statistics involve the sibling correlation (r) and variance (σ^2) parameters. To assess the robustness of the statistics to misspecification of the trait parameters, we first tried

using the sample parameter values for each dataset rather than the known population values. Using sample parameter values does not change the type I error (results not shown). For population samples, as one would expect, using sample parameter estimates also has no effect on power. For selected samples, there is a drastic reduction in power for all statistics and all models (results not shown). This is not surprising, since sample estimates calculated from selected samples are generally quite far off from the correct population values.

We next investigated the effect of misspecifying one parameter at a time. For each run, we set two of the parameters to the population values, and set the third to an arbitrary "wrong guess" (see methods). We performed these sensitivity studies using only models 1 and 1'. Tables 12 through 15 give the power results. (Type I error was not sensitive to parameter misspecification for any of the statistics). For each table, we generated a single set of 1000 datasets and analyzed it under different assumed parameter values. Table 12 shows power results for model 1 with population sampling. Table 13 shows the results for model 1' with population sampling. Tables 14 and 15 give the results for selected sampling.

For the population samples (tables 12 and 13), misspecification of the variance has very little effect. Misspecification of the correlation does reduce power a bit for the group III statistics. Misspecifying the mean substantially reduces the power of the group III statistics, and reduces the power of XU slightly. ORIGINAL.HE, which does not depend on the mean, has roughly equivalent power to XU when the mean is misspecified for model 1, but not for model 1'. Overall, XU seems to be the statistic with the best power in the face of parameter misspecification.

For the selected samples from model 1 (table 14), misspecifying the variance again has very little effect. Misspecifying the correlation decreases the power of the group III statistics

slightly. Misspecifying the mean causes moderate decreases in power; HE-COM-correlation seems to be the most resistant to this effect. ORIGINAL.HE and TRAIT.PRODUCT perform just about as well as the group II and group III statistics when the parameters are misspecified. Overall, HE-COM-correlation, TRAIT.PRODUCT and ORIGINAL.HE look like the most robust statistics in this table. For selected samples from model 1' (table 15), we obtain fairly similar results; HE-COM-correlation again appears to have the most robust power. For this model, however, ORIGINAL.HE and TRAIT.PRODUCT are not as powerful. Note that in some cases misspecification actually increases the power. Presumably this is because the population trait parameters are only optimal under normality assumptions.

We did a limited study of the effects of misspecifying two parameters at a time. Detailed results are not shown, but the general qualitative result was that power was driven by how badly the mean was misspecified. This is consistent with the "one parameter wrong" runs described above, in which the mean had by far the greatest effect on power.

4.5. DISCUSSION

We have performed the most comprehensive comparison to date of statistics for QTL mapping using sibling pairs. We used simulation to evaluate the type I error and power of twelve different statistics under both population and selected sampling. Seven of the statistics (ORIGINAL.HE, TRAIT.SUM, TRAIT.PRODUCT, XU, HE-COM-correlation, HE-COM-combination, and SCORE3) have consistently correct type I error over all the models and sampling schemes we considered. If one considers only the results for perfectly known trait parameters, the statistics with the highest power are XU, HE-COM-correlation, HE-COM-combination, and SCORE3; they are just about equivalent except that XU has higher power for the non-normal trait models

we studied. This suggests that any of those statistics would be appropriate for most studies, with a possible preference for XU depending on the trait distribution. However, when the effect of misspecification of parameters is taken into account, the picture changes somewhat. For population samples, XU appears to have the most robust power, but if one has a population sample one also has decent estimates of the population parameters. Parameter misspecification is a much more important issue for selected samples, and in that case HE-COM-correlation seems most robust.

Our results are basically consistent with those of previous studies. The finding that the group I statistics are not as powerful as the group II statistics was demonstrated previously by a number of different authors (e.g. Xu et al. 2000, Forrest 2001, and Visscher and Hopper 2001). Neither Forrest (2001) nor Visscher and Hopper (2001) observed incorrect type I error for their methods, but they looked only at population samples from a limited number of distributions. The sensitivity of TRAIT.PRODUCT to misspecification of the mean was examined by Palmer et al. (2000) and Zhang et al. (2002). The approximate equivalence of XU, HE-COM-correlation, and HE-COM-combination was noted based on analytical arguments by Sham and Purcell (2001). The similarity between HE-COM-combination and the score statistics was noted (again based on analytical arguments) by Feingold (2001).

There are of course limitations to our study in the types of samples we considered, in the models we considered, and in the statistics we considered. In terms of the types of samples, the most important limitation is that we only considered sibling pairs. Real studies generally include larger sibships as well. Of the more powerful statistics we considered, both XU and the score statistics generalize to larger sibships; HE-COM-correlation and HE-COM-combination do not. It is possible that the different methods for handling larger sibships result in substantial power

differences, so further study of these methods is very important. A method that was developed specifically for extended pedigrees is the regression-based method of Sham et al. (2002), which is discussed further below.

An additional limitation in the types of samples we considered is that we studied one-tailed sampling (one sib in the top 10%), ignoring two-tailed sampling (one sib in the top 10% ***or*** in the bottom 10%). We expect that the statistical performance results for one tail would generally hold for two. Statistics with incorrect type I error for one-tailed sampling will likely be incorrect also for two-tailed sampling. The group of statistics with equal high power for one-tailed sampling is likely to also have highest power for two-tailed sampling. Chapter 5 considers discordant pairs (one sib in the top 10% *and* one in the bottom 10%); for that type of sampling the results are quite different from the ones presented here.

We only considered large sample sizes (large numbers of sibling pairs). We assume that studies with small samples are fairly unusual.

There are also some limitations of the models we studied. All of our models used an environmental/polygenic sibling correlation of .25. This should not affect the relative power of the group II and group III statistics. It does affect the relative power of the group I statistics, but since this fact is well-documented (Palmer et al. 2000, Forrest 2001), we did not explore it in detail here. It does mean that the relative power of ORIGINAL.HE and TRAIT.PRODUCT that we observed should not be taken as a general rule. In general, the greater the correlation, the better ORIGINAL.HE performs in comparison to TRAIT.PRODUCT. We do see a need for further exploration of a wide variety of non-normal models. The fact that our results for models 1' and 2' were significantly different from our results for mixture-of-normals models indicates

the need for further work, preferably with careful consideration given to what types of models are realistic.

There are several important statistics in the literature that we did not include in our study. We did not consider variance components because our main focus was on selected sampling, and it is well-documented that variance components has incorrect type I error in such cases (e.g. Allison et al. 1999, Sham et al. 2000). There are, however, various robust versions of variance components (see Feingold 2001 for a review) that could be more carefully compared to the methods discussed here. We also did not consider the precise score statistics proposed by Wang and Huang (2002a) and Putter et al. (2002). Given that SCORE3 performs very well, further study of other score statistic variations might be useful. The statistic proposed by Sham et al. (2002), mentioned above, was developed specifically to be a robust statistic for extended pedigrees. It regresses IBD on trait values, the opposite of the statistics discussed in this study. For sibling pairs it has exactly the same form as SCORE2 and SCORE3, but with the variance of π in the denominator estimated differently.

Finally, there are a few potentially useful variations on the statistics we considered that are not yet in the literature. One could use any of the statistics here with the parameter estimates chosen (based on the data) to maximize the value of the statistic. This is particularly appealing as a way to deal with the sensitivity to the mean. Similarly, one could use a statistic that weights the squared sum and squared difference regressions, like XU, but with the weights maximized for the particular dataset. Either of these approaches would entail the loss of a degree of freedom to properly adjust for the maximization, and we suspect that as a result there would not be a useful power gain, but further study is warranted.

Table 7 Genetic models –chapter 4

	Models										
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1'**	**2'**
Type of inheritance	Add	Dom	Rec	Add	Dom	Rec	Add	Dom	Rec	Add	Dom
Locus heritability	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	N.A.	N.A.
Allele frequency	0.1	0.1	0.1	0.5	0.5	0.5	0.9	0.9	0.9	0.1	0.1
Trait means	-1, 0, 1	0, 1, 1	0, 0, 1	-1, 0, 1	0, 1, 1	0, 0, 1	-1, 0, 1	0, 1, 1	0, 0, 1	-1.6, 0, 1.6	0, 1.6, 1.6
Env. std. dev.	0.849	0.785	0.199	1.414	0.866	0.866	0.849	0.199	0.785	N.A.	N.A.
Env. correlation	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	N.A.	N.A.
Overall mean	-0.8	0.19	0.01	0.0	0.75	0.25	0.8	0.99	0.81	-1.32	0.295
Overall std. dev.	0.949	0.877	0.222	1.581	0.968	0.968	0.949	0.222	0.877	2.047	1.393
Skewness	0.168	0.140	0.880	0.0971	-0.0991	0.102	-0.168	-0.880	-0.140	-1.587	1.504
Kurtosis	0.101	0.0240	3.802	0.0556	-0.0714	-0.031	0.101	3.802	0.0240	5.268	9.406
Overall Corr.	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.25	0.26

Table 8 Type I error for population samples

	Model									
	1		2		3		1'		2'	
	SD	Error (%)	SD	Error (%)	SD	Error (%)	SD	Error (%)	SD	Error (%)
Group I										
ORIGINAL.HE	1.01	1.17	1.01	1.07	0.99	1.00	1.00	1.02	1.00	1.12
TRAIT.SUM	1.01	1.16	0.99	1.03	0.99	0.94	1.01	1.05	1.00	0.84
TRAIT.PRODUCT	1.02	1.16	1.00	1.00	0.99	0.88	1.00	1.03	1.00	0.86
Group II										
XU	1.01	1.12	1.01	1.03	0.99	0.89	1.00	0.89	0.99	0.94
V&H	1.01	1.12	1.01	0.99	0.85	0.26	0.72	0.12	0.63	0.01
FORREST	1.01	1.12	1.01	1.02	0.86	0.34	0.75	0.15	0.69	0.04
Group III										
HE-COM-correlation	1.01	1.13	1.01	1.05	0.99	0.94	1.00	1.01	1.00	0.93
HE-COM-combination	1.02	1.12	1.01	1.04	0.99	0.93	1.00	1.06	1.00	0.93
SCORE1	0.96	0.70	1.01	0.76	1.29	3.53	1.10	1.87	3.26	23.00
SCORE2	0.94	0.65	1.01	0.63	0.92	0.58	0.92	0.55	0.93	0.50
SCORE3	1.01	1.10	1.01	1.00	0.99	0.92	0.99	1.06	1.00	0.90
SCORE4	1.01	1.12	1.01	1.03	0.99	0.94	1.00	0.99	0.98	0.84

Note- the nominal type I error was set at 1%

Table 9 Type I error for selected samples

	Model									
	1		2		3		1'		2'	
	SD	Error (%)	SD	Error (%)	SD	Error (%)	SD	Error (%)	SD	Error (%)
Group I										
ORIGINAL.HE	1.00	1.00	1.00	0.90	1.01	1.18	1.01	0.97	1.00	0.75
TRAIT.SUM	1.00	0.89	1.00	1.10	0.99	0.79	1.02	1.00	1.00	1.03
TRAIT.PRODUCT	1.00	0.90	1.01	1.00	0.99	0.84	1.02	1.00	1.00	0.98
Group II										
XU	1.00	1.01	1.00	0.97	1.00	1.05	1.01	0.93	0.99	0.98
V&H	1.11	1.66	1.05	1.20	0.90	0.55	0.87	0.35	0.75	0.08
FORREST	1.13	1.83	1.15	2.10	0.91	0.60	0.91	0.41	0.79	0.13
Group III										
HE-COM-correlation	1.00	0.94	1.00	1.00	1.00	1.03	1.01	0.93	0.99	0.84
HE-COM-combination	1.00	1.04	1.01	1.20	1.00	1.05	1.02	1.10	0.99	0.85
SCORE1	1.62	7.21	1.90	11.00	2.75	19.48	3.25	24.00	7.63	37.54
SCORE2	0.93	0.64	0.93	0.76	0.92	0.68	0.94	0.63	0.92	0.50
SCORE3	1.00	0.98	1.00	1.20	0.99	1.01	1.01	1.10	0.99	0.80
SCORE4	0.99	0.87	0.86	0.34	0.98	0.88	0.83	0.17	0.93	0.48

Note- the nominal type I error was set at 1%

Table 10 Power for population samples at α=1% level

	Model										
	1	2	3	4	5	6	7	8	9	1’	2’
Group I											
ORIGINAL.HE	0.61	0.60	0.23	0.59	0.57	0.57	0.58	0.22	0.56	0.09	0.15
TRAIT.SUM	0.15	0.19	0.07	0.19	0.19	0.20	0.16	0.07	0.16	0.03	0.04
TRAIT.PRODUCT	0.53	0.55	0.27	0.54	0.55	0.57	0.53	0.25	0.51	0.15	0.36
Group II											
XU	0.71	0.73	0.39	0.72	0.72	0.70	0.71	0.37	0.69	0.21	0.54
Group III											
HE-COM-correlation	0.71	0.73	0.42	0.72	0.71	0.70	0.71	0.40	0.69	0.21	0.44
HE-COM-combination	0.71	0.73	0.41	0.72	0.71	0.70	0.71	0.40	0.69	0.21	0.43
SCORE3	0.71	0.73	0.41	0.72	0.71	0.69	0.71	0.40	0.69	0.21	0.43

Table 11 Power for selected samples at α=1% level

	Model										
	1	2	3	4	5	6	7	8	9	1’	2’
Group I											
ORIGINAL.HE	0.84	0.82	0.72	0.58	0.33	0.75	0.29	0.04	0.28	0.39	0.23
TRAIT.SUM	0.61	0.66	0.24	0.35	0.15	0.55	0.10	0.01	0.12	0.17	0.18
TRAIT.PRODUCT	0.85	0.87	0.76	0.57	0.31	0.80	0.23	0.03	0.23	0.70	0.62
Group II											
XU	0.89	0.88	0.92	0.64	0.36	0.83	0.29	0.02	0.31	0.77	0.75
Group III											
HE-COM-correlation	0.90	0.89	0.93	0.65	0.36	0.84	0.28	0.03	0.31	0.73	0.65
HE-COM-combination	0.91	0.91	0.97	0.66	0.37	0.86	0.28	0.03	0.31	0.69	0.68
SCORE3	0.91	0.91	0.94	0.65	0.37	0.85	0.28	0.03	0.30	0.69	0.67

Table 12 Power for population samples: sensitivity analyses under model 1

	$r = 0.1$	$r = 0.5$	$\mu = -1.75$	$\mu = 0.15$	$\sigma^2 = 0.45$	$\sigma^2 = 1.8$	*Population*
Group I							
ORIGINAL.HE	0.61	0.61	0.61	0.61	0.61	0.61	0.61
TRAIT.SUM	0.15	0.15	0.04	0.05	0.15	0.15	0.15
TRAIT.PRODUCT	0.53	0.53	0.15	0.17	0.53	0.53	0.53
Group II							
XU	0.71	0.71	0.65	0.63	0.71	0.71	0.71
Group III							
HE-COM-correlation	0.63	0.69	0.49	0.47	0.71	0.71	0.71
HE-COM-combination	0.61	0.65	0.41	0.42	0.71	0.67	0.71
SCORE3	0.61	0.65	0.41	0.42	0.71	0.67	0.71

Table 13 Power for population samples: sensitivity analyses under model 1'

	r = 0.1	r = 0.5	μ = -3.37	μ = 0.73	σ^2 = 2.095	σ^2 = 8.38	*Population*
Group I							
ORIGINAL.HE	0.09	0.09	0.09	0.09	0.09	0.09	0.09
TRAIT.SUM	0.03	0.03	0.04	0.02	0.03	0.03	0.03
TRAIT.PRODUCT	0.15	0.15	0.09	0.05	0.15	0.15	0.15
Group II							
XU	0.21	0.21	0.11	0.18	0.21	0.21	0.21
Group III							
HE-COM-correlation	0.20	0.13	0.12	0.15	0.21	0.21	0.21
HE-COM-combination	0.19	0.12	0.11	0.13	0.20	0.20	0.21
SCORE3	0.19	0.12	0.11	0.12	0.20	0.20	0.21

Table 14 Power for selected samples: sensitivity analyses under model 1

	$r = 0.1$	$r = 0.5$	$\mu = -1.75$	$\mu = 0.15$	$\sigma^2 = 0.45$	$\sigma^2 = 1.8$	*Population*
Group I							
ORIGINAL.HE	0.84	0.84	0.84	0.84	0.84	0.84	0.84
TRAIT.SUM	0.61	0.61	0.64	0.14	0.61	0.61	0.61
TRAIT.PRODUCT	0.85	0.85	0.80	0.91	0.85	0.85	0.85
Group II							
XU	0.89	0.89	0.79	0.78	0.89	0.89	0.89
Group III							
HE-COM-correlation	0.88	0.88	0.88	0.89	0.90	0.90	0.90
HE-COM-combination	0.83	0.87	0.63	0.85	0.91	0.90	0.91
SCORE3	0.82	0.87	0.62	0.85	0.91	0.89	0.91

Table 15 Power for selected samples: sensitivity analyses under model 1'

	r = 0.1	r = 0.5	μ = -3.37	μ = 0.73	σ^2 = 2.095	σ^2 = 8.38	*Population*
Group I							
ORIGINAL.HE	0.39	0.39	0.39	0.39	0.39	0.39	0.39
TRAIT.SUM	0.17	0.17	0.32	0.00	0.17	0.17	0.17
TRAIT.PRODUCT	0.70	0.70	0.59	0.31	0.70	0.70	0.70
Group II							
XU	0.77	0.77	0.70	0.27	0.77	0.77	0.77
Group III							
HE-COM-correlation	0.76	0.54	0.74	0.50	0.73	0.73	0.73
HE-COM-combination	0.63	0.58	0.21	0.52	0.73	0.58	0.69
SCORE3	0.63	0.57	0.21	0.52	0.72	0.58	0.69

5. RECENT ADVANCES IN HUMAN QUANTITATIVE-TRAIT-LOCUS MAPPING: COMPARISON OF METHODS FOR DISCORDANT SIBLING PAIRS

This chapter has been published in *American Journal of Human Genetics* (Szatkiewicz et al. 2003). I have obtained the copyright permission from The Chicago Press. The content of Szatkiewicz et al. (2003) is used below without change but its format has been modified to fit this dissertation. I wrote the simulation program, co-authored the statistical program, and conducted all simulation studies for that paper. Dr. Karen T.Cuenco co-authored the statistical program.

5.1. SUMMARY

Extreme discordant sibling pairs (EDSPs) are theoretically powerful for mapping quantitative trait loci in humans. EDSPs have not been used much in practice, however, because of the need to screen very large populations in order to find enough pairs that are extremely discordant. Given appropriate statistical methods, another alternative is moderately discordant sibling pairs (MDSPs) – pairs that are discordant but not at the far extremes of the distribution. Such pairs can be powerful, yet far easier to collect than extreme discordant pairs. Recent work on statistical methods for quantitative trait locus mapping in humans has included a number of methods that, while not specifically developed for discordant pairs, may well be powerful for MDSPs and possibly even EDSPs. In this chpater, we survey the new statistics and discuss their applicability to discordant pairs. We then use simulation to study the type I error and power of various

statistics for EDSPs and for MDSPs. We conclude that the best statistic(s) for discordant pairs (moderate or extreme) are to be found among the new statistics. We suggest that the new statistics are appropriate for many other designs as well, and in fact that they open the way for exploration of entirely novel designs.

5.2. INTRODUCTION

The extreme discordant sibling pair (EDSP) design is generally attributed to Risch and Zhang (1995). The basic idea of that design is that if phenotyping is relatively easy, one can screen a large population of sibling pairs and genotype only those pairs that are most powerful for detecting linkage. The simplest version of the design uses only those pairs in which one sibling has a trait value in the top 10% of the trait distribution and the other sibling has a trait value in the bottom 10% of the trait distribution. The EDSP idea was further developed in work such as that of Risch and Zhang (1996), Gu et al. (1996), Kruse et al. (1997), Rogus et al. (1997), and Knapp (1998). These authors studied the power of several variations on EDSP sampling, including the extreme discordant and concordant (EDAC) design, which includes pairs where both siblings are in the top 10% or both are in the bottom 10%. Despite theoretical development, EDSP and EDAC designs have only occasionally been used in practice. Only investigators with very large populations to work with and relatively low phenotyping costs have found such studies to be practical. For example, Xu et al. (1999) screened over 200,000 people in Anqing, China in order to ascertain 207 extreme discordant and 357 extreme concordant pairs for blood pressure. Fullerton et al. (2003) screened 20,427 independent sibships to get a final dataset of 182 discordant and 379 concordant pairs for neuroticism.

The reason for the historical emphasis on *extreme* discordant pairs has to do with the statistical test that is used to detect linkage. In a standard EDSP study, one tests for linkage by estimating the average number of alleles shared identical by descent (IBD) between the pairs at a marker. If the marker locus is linked to the trait, the mean IBD sharing score should be less than the null hypothesis expectation. This is the same test statistic (tested in the opposite direction) that is used for affected sibling pair mapping of binary disease traits (e.g., Blackwelder and Elston 1985). But, the IBD sharing statistic is not very powerful unless the pairs are drawn from the extremes of the distribution. This is because the power of the statistic comes purely from the fact that the trait values are extreme; it does not use the actual trait values in any way.

More recently, the suggestion has been made that one can use different statistics for discordant pairs – statistics that use information about the IBD sharing but also incorporate information about the trait values. Such statistics can make it possible to use less extreme samples and might make discordant pair studies more practical. Forrest and Feingold (2000) suggested using a composite statistic that is a weighted sum of the IBD sharing statistic and the Haseman and Elston (1972) statistic. The composite statistic is only slightly more powerful than the IBD sharing statistic for EDSP samples, but the advantage is greater for moderately discordant sibling pairs, defined arbitrarily by Forrest and Feingold as pairs with one sibling in the top 35% of the distribution and one sibling in the bottom 35% of the distribution. The existence of a powerful statistic for MDSPs makes it possible to consider that design as a compromise between EDSPs and population sampling. For example, under one trait model that Forrest and Feingold studied, one could achieve 80% power by screening 8700 pairs to ascertain 55 EDSPs, or by screening 1850 pairs to ascertain 300 MDSPs, or by using a population sample of 950 pairs.

In addition to the Forrest and Feingold composite statistic, there are several other recent methods that may also be applicable to discordant pairs. In this chapter, we survey those statistics, and then use simulation to compare their type I error and power to those of the IBD sharing statistic. In the methods section, we discuss statistics for discordant pairs in more detail and define the statistics we consider in this chapter. We then describe our simulation methods and present our results. We conclude with a discussion of the implication of our results for other study designs.

5.3. METHODS

5.3.1. Statistics for discordant sibling pairs

Discordant and concordant pairs have a property that makes them critically different from more typical samples – they have a distorted IBD sharing distribution at markers that are linked to the trait. A population sample of sibling pairs is expected to share half of their alleles IBD at any locus, regardless of whether that locus is linked to the trait being studied. The same is true if the pairs are sampled on the basis of a single individual with an extreme trait value. But if families are sampled based on a criterion that looks at two or more members (which we refer to as multiple-proband ascertainment), then the IBD distribution changes at a marker linked to the trait. In the companion paper, we sampled pairs in which *at least one* sibling exceeds a threshold; this actually qualifies as multiple-proband ascertainment (because both phenotypes must be seen in order to decide whether to ascertain the pair), but only changes the IBD distribution *very* slightly. In the case of discordant and concordant pairs, the change in the IBD distribution is quite substantial, and this is exactly what Risch and Zhang's original EDSP method sought to detect.

But the IBD-sharing statistic for EDSP pairs is limited in that it *only* looks at the IBD sharing, ignoring the actual trait values.

Conversely, most standard human QTL-mapping methods, which were developed with population samples in mind, do not look at the marginal distribution of IBD sharing at all. They base their power on detecting correlation between the IBD sharing of each pair and the similarity of the pair's trait values. We refer to such statistics for the remainder of this chapter as correlation-based statistics. Haseman-Elston regression (1972) and maximum-likelihood variance components (e.g. Amos 1994) are the most commonly-used examples of correlation-based methods.

The set of correlation-based statistics has recently expanded quite a bit, with the attempts to "update" the Haseman-Elston method (Drigalenko 1998, Elston et al. 2000, Xu et al. 2000, Forrest 2001, Visscher and Hopper 2001, Sham and Purcell 2001). There are also several new statistics in the literature that combine information from both the marginal IBD sharing distribution and the correlation (Sham et al. 2000, Sham and Purcell 2001, Tang and Siegmund 2001, Wang and Huang 2002, Putter and Sandkuijl 2002, Sham et al. 2002). These statistics should be appropriate for discordant pairs, and they should be more powerful than statistics that rely only on IBD or only on correlation. The new statistics are described briefly below, and are reviewed in more detail in Feingold (2001) and Feingold (2002). The statistics that we designate as "group A" (IBD1, IBD2) are versions of the traditional IBD sharing statistic. The "group B" statistics (ORIGINAL.HE, TRAIT.SUM, TRAIT.PRODUCT, XU, V&H, FORREST, HE-COM-correlation, SCORE4) are the correlation-based statistics. They do not consider the marginal IBD sharing distribution. The "group C" statistics (HE-COM-combination, SCORE1, SCORE2, SCORE3, COMPOSITE1, COMPOSITE2) are the statistics that consider both IBD sharing and correlation.

Risch and Zhang IBD- sharing statistic (IBD1). Let π_i be the estimated mean IBD sharing for sibling pair *i*; π_i takes values 0, 1/2, or 1 for a fully-informative pair, but can take intermediate values if multi-point estimates are used. Let $\overline{\pi}$ be the average estimated IBD sharing over all pairs in the sample. The classical linkage test using EDSPs (Risch and Zhang 1995) uses the statistic

$$\frac{\overline{\pi} - 1/2}{\sqrt{\frac{1}{8n}}},$$

which is $\overline{\pi}$ standardized to have mean 0 and variance 1. A one-sided Z-test is used to detect significantly negative values. The standard deviation in the denominator is a theoretical value that assumes IBD information for each pair is observed perfectly (i.e. that the marker is infinitely polymorphic). This results in a conservative test when this statistic is applied to real data in which IBD sharing is estimated from marker data (see Davis and Weeks 1997 for a discussion of this issue in the context of affected sibling pairs).

Robust IBD-sharing statistic (IBD2). Instead of the denominator used above, the IBD-sharing statistic can also be standardized using an empirical standard deviation, which yields the statistic

$$\frac{\overline{\pi} - 1/2}{\sqrt{\sum (\pi_i - 1/2)^2 / n^2}}.$$

The test based on this statistic should have correct type I error even if the IBD information is not perfectly observed. One could also consider replacing the factor of 1/2 in the denominator with $\overline{\pi}$, which (named as *IBD3*)should result in a very slightly elevated type I error and power.

Original Haseman-Elston (ORIGINAL.HE). Let $Y_{iD} = (x_{i1}\text{-}x_{i2})^2$ be the squared trait difference for sibling pair *i*. The method of Haseman and Elston (1972) simply regresses Y_{iD} on π_i and estimates the slope, $-\beta_D$. A positive estimate for β_D (a negative estimate for the slope) suggests that the trait is linked to the locus marker. A one-sided t-test is used to test for any significant departure from zero.

Trait-sum regression (TRAIT.SUM). Let $Y_{iS} = [(x_{i1} - \mu) + (x_{i2} - \mu)]^2$ be the mean-corrected squared trait sum. We include the one-sided t-test of the slope β_S from the regression of Y_{iS} on π_i.

Trait-product regression (TRAIT.PRODUCT). Under population sampling, β_D and β_S are estimates of the same parameter (Drigalenko 1998). This slope parameter should be zero under the null hypothesis of no linkage, and should be positive (as we have defined the sign) under the alternative hypothesis. Drigalenko (1998) suggested averaging the two slope estimates, or , equivalently, doing a single regression with the mean-corrected trait product, $[(X_{i1} - \mu)(X_{i2} - \mu)]$, as the dependent variable. We consider the one-sided t-test based on the trait-product regression.

Forrest's method (FORREST). Forrest (2001) suggested a test based on the weighted average

$$\hat{\beta} = \frac{\sigma_D^2}{\sigma_D^2 + \sigma_S^2}\hat{\beta}_S + \frac{\sigma_S^2}{\sigma_D^2 + \sigma_S^2}\hat{\beta}_D \ ,$$

where σ_D^2 and σ_S^2 are the variances of $\hat{\beta}_D$ and $\hat{\beta}_S$. These weights are optimal assuming that the covariance, σ_{DS}^2, of $(\hat{\beta}_D, \hat{\beta}_S)$ is zero, which is true for a population sample from a normal distribution, but not necessarily otherwise (Feingold 2002). Forrest's method estimates all the parameters simultaneously using iterative least squares.

Visscher and Hopper's method (V&H). Visscher and Hopper (2001) proposed a test based on the same weighted slope estimate as Forrest (2001), but with the two variances estimated separately by performing the two regressions separately.

Xu et al.'s method (XU). Xu et al. (2000) proposed a method very similar to that of Forrest (2001) and Visscher and Hopper (2001), but their weighted average slope allows for a non-zero covariance between $\hat{\beta}_D$ and $\hat{\beta}_S$ using the formula

$$\hat{\beta} = \frac{\sigma_S^2 - \sigma_{DS}^2}{\sigma_D^2 + \sigma_S^2 - 2\sigma_{DS}^2}\hat{\beta}_D + \frac{\sigma_D^2 - \sigma_{DS}^2}{\sigma_D^2 + \sigma_S^2 - 2\sigma_{DS}^2}\hat{\beta}_S.$$

Xu et al. estimate the parameters by performing the two regressions separately, similarly to Visscher and Hopper. The covariance can be estimated by combining the residuals of the two regressions.

Sham and Purcell's method (HE-COM-correlation). The variances σ_D^2 and σ_S^2 can actually be calculated analytically as functions of the sibling trait correlation, *r*, under traditional QTL models. Sham and Purcell (2001) proposed taking advantage of this fact, rather than estimating the variances from data as in FORREST, V&H, and XU. The primary method outlined in Sham and Purcell (2001) regresses the dependent variable

$$\frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2}$$

on π_i , where the trait values x_{i1} and x_{i2} are standardized to mean zero and variance one before calculating Y_{iS} and Y_{iD}.

Sham and Purcell's robust method (HE-COM-combination). Sham and Purcell also suggested a variant of their method, regressing

$$A_i = \frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2} + \frac{4r}{1-r^2}$$

on π_i - 1/2, with the intercept fixed at zero. This variant should be more robust to selected sampling. Even more importantly, it implicitly incorporates information on any distortion in the IBD sharing. This is because the fixed intercept implies a null hypothesis IBD sharing proportion of 1/2, and so the regression t-test draws power from any deviation from that proportion. The t-statistic for the test of the regression slope is

$$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left\{\left[\sum A_i^2\right]\left[\sum(\pi_i - 1/2)^2\right] - \left[\sum A_i(\pi_i - 1/2)\right]^2\right\}}}.$$

Asymptotic score statistic (SCORE1). Score statistics based on the usual variance components likelihood were proposed by Tang and Siegmund (2001), Wang and Huang (2002), and Putter et al. (2002). The score statistics proposed in these papers are very similar to each other, but have minor differences in how they parameterize the likelihood and how they

"robustify" the statistic. Instead of considering precisely the statistics in these papers, we take the Tang and Siegmund (2001) statistic as our starting point and consider four variations on possible ways to make it robust (or not). Tang and Siegmund (2001) derived a score statistic of the form

$$\frac{\sum_i A_i(\pi_i - 0.5)}{\sqrt{2n\left[\frac{1+r^2}{(1-r^2)^2}\right]}},$$

where A_i is the same function defined for HE-COM-combination above. The denominator of this statistic is based on asymptotic likelihood theory, so this version of the score statistic is not expected to be appropriate for discordant sibling pairs.

Score statistic with partially empirical variance (SCORE2). Tang and Siegmund (2001) proposed making their statistic robust by using the empirical standard deviation of A_i in the denominator, i.e.

$$\frac{\sum_i A_i(\pi_i - 0.5)}{\frac{1}{2\sqrt{2}}\sqrt{\sum_i A_i^2}}.$$

The factor of $1/2\sqrt{2}$ is the standard deviation of π assuming a perfectly informative marker. Thus, this version of the statistic should be appropriate for discordant pairs, but should yield a conservative test when there is imperfect IBD information.

Score statistic with fully empirical variance (SCORE3). We propose that the best version of the score statistic should have the same form as SCORE2, but with the empirical standard deviation of π in place of the factor of $1/2\sqrt{2}$:

$$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - 1/2)^2\right]}}.$$

This version should have correct type I error even with imperfect IBD information. As with IBD2, it is also possible here to replace the factor of 1/2 in the denominator with $\bar{\pi}$, which (named as *SCORE5*) would give slightly higher type I error and power.

Score statistic with empirical mean and variance (SCORE4). Both Wang and Huang (2002a) and Putter et al. (2002) proposed using $\bar{\pi}$ in place of 1/2 in both the *numerator and denominator* of the score statistic. Applied to our parameterization of the score statistic, that yields the expression

$$\frac{\sum A_i(\pi_i - \bar{\pi})}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - \bar{\pi})^2\right]}}.$$

The use of the empirical mean IBD sharing *in the numerator* means that this version of the score statistic *does not* draw any power from the distortion in IBD sharing in discordant pairs, similarly to HE-COM-correlation (while SCORE3 is similar to HE-COM-combination).

Composite statistic (COMPOSITE1). Forrest and Feingold (2000) proposed to test for linkage using a weighted average of ORIGINAL.HE and IBD1:

$$w_{HE}\frac{-\hat{\beta}_D}{\hat{\sigma}_D} + w_{IBD}\frac{\bar{\pi} - 1/2}{\sqrt{1/(8n)}},$$

where w_{HE} and w_{IBD} are arbitrarily-chosen weights. Based on limited calculations they recommended analyzing MDSPs using equal weights and EDSPs using a higher weight on the IBD-sharing statistic. We use equal weights in this chapter. Any of the group B statistics can be used in place of ORIGINAL.HE in the composite. Forrest and Feingold found that for discordant pairs ORIGINAL.HE was the most powerful choice in the literature at that time. Updating that investigation with all of the group B statistics (peek ahead to tables 13 and 14), we found that ORIGINAL.HE was still the most powerful, so we implemented the composite statistic as above. Note that the IBD-sharing component of the composite statistic is standardized using the theoretical variance, so it is expected to be conservative when there is imperfect IBD-sharing information.

Empirical composite statistic (COMPOSITE2). The composite statistic can also be formed as the average of ORIGINAL.HE and IBD2 (instead of IBD1). This version should have correct type I error even when there is not perfect IBD information.

We are aware of two other statistics that fall into group C (combining IBD and correlation information), but that we did not include in our study due to computational limitations. One is the ascertainment-corrected variance components statistic proposed by Sham et al. (2000), which conditions on trait values. That statistic should perform very similarly to SCORE3 and HE-COM-combination. The other statistic that we did not include is the regression-based statistic proposed by Sham et al. (2002). This statistic was developed for extended pedigrees, but for sibling pairs it takes exactly the same form as SCORE2 and SCORE3, except that the variance of π is estimated differently.

Many of the statistics we evaluate depend on estimates of trait parameters. The statistics TRAIT.SUM, TRAIT.PRODUCT, XU, V&H, FORREST, HE-COM-correlation, HE-COM-combination, SCORE1, SCORE2, SCORE3, and SCORE4 all use an estimate of the trait mean, μ. The S&P statistics and the SCORE statistics additionally use estimates of the trait variance, σ^2, and the sibling correlation, r. Sensitivity to these estimates may have an important effect on power.

5.3.2. Simulations

We studied the type I error and power of each statistic under seven trait models, which are described in table 16. All models are diallelic. Models one through five are simple mixture-of-normals models; the trait value is equal to the genotype mean plus a normally-distributed "environmental" variance. There is an additional sibling correlation of .25 in each model to account for environmental and polygenic components. The means and variances were chosen to give each model a locus heritability of 0.2. Models 1' and 2' were generated by simulating data under models 1 and 2, respectively, and then taking the signed square ($x|x|$) of each trait value. This yields overall trait distributions that are somewhat skewed and have high kurtosis. Model 3 also has skewness and kurtosis in the same range as models 1' and 2'. Note that models 6 through 9 from the companion paper were not used here because for discordant pair sampling they are symmetric with other models.

We simulated data for nuclear families with two children according to each of the models, and ascertained families by two different methods. The first ascertainment scheme was EDSPs – any pair in which one sibling was in the top 10% of the trait distribution and the other sibling was in the bottom 10%. The second scheme was MDSPs – any pair in which one sibling was in the top 35% and the other was in the bottom 35%. We simulated datasets of 100 families for the

EDSP samples and 500 families for the MDSP samples. Figure 3 shows examples of simulated bivariate trait distributions for both sampling schemes under several of the models. To study type I error, we used 10,000 datasets, and to study power we used 1,000 datasets. The nominal type I error rate was set at .01. Marker data was simulated using eight equally-frequent alleles, with the marker at recombination fraction $\theta = 0$ for the power study and $\theta = 1/2$ for the type I error study. We also did power simulations at $\theta = 0.05$ for models 1 and 2 only.

As discussed above, most of the statistics require that some trait parameters (mean μ, variance σ^2, sibling correlation r) be specified. In general, theory suggests that these should be population parameter values, even for selected samples. However, if one is using a selected sample, population parameter estimates may not be available. In that situation parameter values must be guessed or adopted from previous studies in other populations. We examined the robustness of the statistics to misspecification of parameters using models 1 and 1' only. We varied one parameter at a time while holding the other two parameters at the correct population values. Sibling correlation was set at 0.1 and 0.5, trait variance at values ranging from half the true value to twice the true value and trait mean at the true mean plus and minus one standard deviation. We also did a limited number of studies with two parameters at a time misspecified. Finally, we checked the performance of the statistics using sample estimates of the parameters.

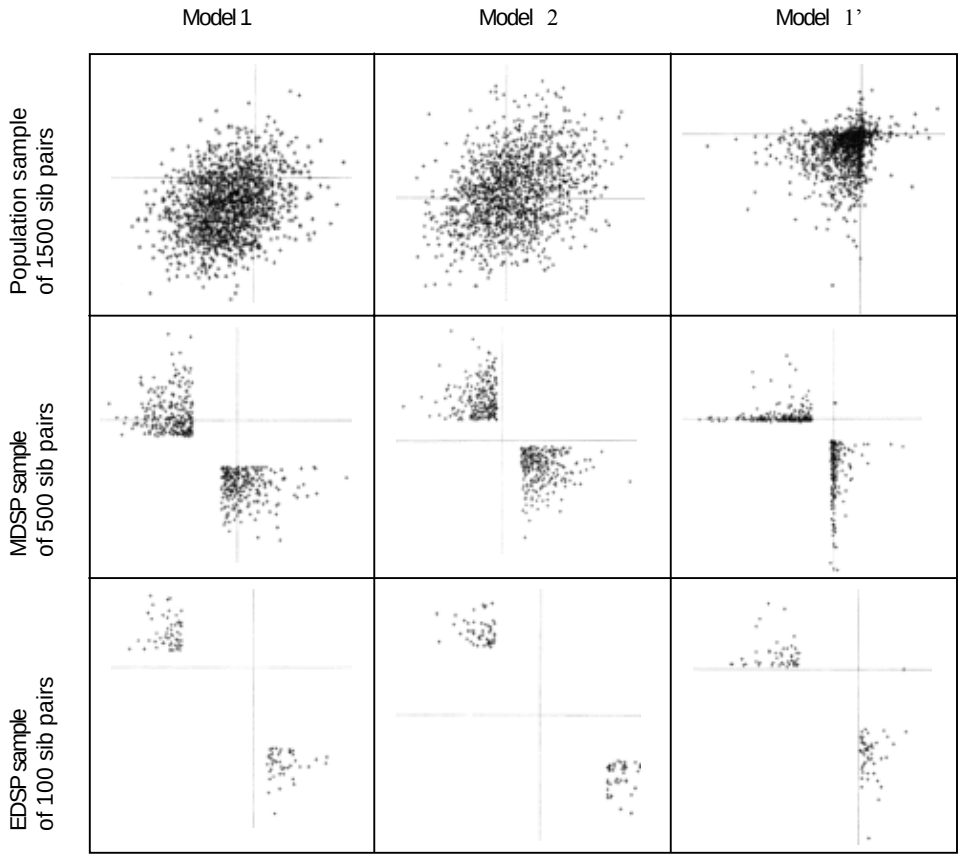

Figure 3 Scatter plots of population samples, MDSP samples, and EDSP samples from models 1, 2, and 1'.

5.4. RESULTS

5.4.1. Type I error

Table 17 shows the standard deviation and type I error of each statistic based on the 10,000 simulated datasets with EDSP samples. All statistics had mean zero for all models. All of the statistics in this table were computed with the known population values of the parameters (trait mean μ, variance σ^2, and sibling correlation r). The 95% confidence interval for an estimated error rate of 1.00% is approximately 0.80% to 1.20%. As expected, the statistics that assume perfect IBD information (IBD1, SCORE2, COMPOSITE 1) have conservative type I error. The statistics V&H and FORREST also have incorrect type I error, presumably due to the omission of the covariance term in the weighting. Finally, SCORE1 and SCORE4 have incorrect type I error. These results are qualitatively consistent across all models, and are also true for the MDSP samples (results not shown). The incorrect type I error for SCORE4 is due to covariance terms that are omitted from the denominator of that statistic (see companion paper for more complete discussion). HE-COM-combination and SCORE3 are very similar, with HE-COM-combination having a slightly higher type I error rate for most models. We did limited experiments (results not shown) with versions of SCORE3, IBD2, and COMPOSITE2 that replaced the "1/2" in the denominator by $\overline{\pi}$ (see methods). This increases the type I error of those methods by 0.1 to 0.2 percentage points for the models we looked at.

5.4.2. Power

Table 18 gives the power for all models for the EDSP samples, and table 19 gives the power for the MDSP samples. Again, all of the statistics in these tables were computed with the known population values of the parameters. To make comparisons simpler, the statistics that did not have

correct type I error are omitted from the power tables. The 95% confidence interval for a power estimate of 50% is approximately 47% to 53%.

For the EDSP samples it is clear that most of the linkage information is in the marginal IBD distribution, with only a small amount in the correlation between IBD and trait differences. The group B statistics, which rely only on correlation, have very little power. IBD2 and the group C statistics all have very similar power. COMPOSITE2 has somewhat lower power than the other group C statistics, because we computed it using equal weights on the IBD statistic and the Haseman-Elston statistic. If COMPOSITE2 were computed with the weights suggested by Forrest and Feingold (2000) for EDSPs, it would probably have similar power to HE-COM-combination and SCORE3. It is interesting to note that HE-COM-combination and SCORE3 do have slightly higher power than IBD2, except against models 1' and 2', for which they have slightly lower power. This suggests that the most "nonparametric" statistic may do better against non-normal trait models.

For the MDSP samples, the group C statistics are again the most powerful. In this case COMPOSITE2 performs very similarly to HE-COM-combination and SCORE3, presumably because the weighting we used to form the composite is in fact the weighting that Forrest and Feingold recommended for MDSPs. IBD2 has somewhat lower power, reflecting the fact that the group C statistics are drawing power from both the marginal IBD distribution and the IBD/trait-difference correlation. COMPOSITE2 outperforms HE-COM-combination and SCORE3 precisely on the non-normal models.

We did limited experiments (results not shown) with versions of SCORE3, IBD2, and COMPOSITE2 that replaced the “1/2” in the denominator by $\overline{\pi}$ (see methods). The altered SCORE3 has power very similar to that of HE-COM-combination. The altered IBD2 and

COMPOSITE2 statistics also gain one to two percentage points in power. We also did power simulations at $\theta = 0.05$ for models 1 and 2 only (results not shown). While the overall power is lower than at $\theta = 0$, the relative power of the different statistics is unchanged.

5.4.3. Sensitivity

To assess the robustness of the statistics to misspecification of the trait parameters, we first tried using the sample parameter values for each dataset rather than the known correct values. The basic effect for both EDSP and MDSP samples is to cut the power of the HE-COM-correlation, HE-COM-combination, and SCORE3 statistics to zero (results not shown). IBD2, ORIGINAL.HE, and COMPOSITE2 do not use the parameter values at all, so they are unaffected.

A more realistic sensitivity analysis is to use parameter values that are guessed with error. We investigated the effect of misspecifying one parameter at a time. For each run, we set two of the parameters to the population values, and set the third to an arbitrary "wrong guess" (see methods). We performed these simulations on the same two datasets, one from model 1 and one from model 1'. Tables 20 through 23 present these results. For each table, we generated a single set of 1000 datasets and analyzed it under different assumed parameter values. Table 20 shows power results for model 1 under EDSP sampling. Table 21 shows the results for model 1' under EDSP sampling. Tables 22 and 23 give the corresponding results for MDSP sampling. The type I error was correct for all of these sensitivity studies (results not shown).

For the EDSP samples from model 1 (table 20), misspecification of the trait parameters has no significant effect on power of the group C statistics. This is because almost all of the power is coming from the IBD-sharing information, which does not depend on the parameters. Under model 1' (table 21) the power is slightly more sensitive to parameter misspecification. This does not affect IBD2 and COMPOSITE2, because they do not use the parameter estimates.

Note that in some cases misspecification actually increases power; this is presumably because the population trait value is only the optimal choice under normality assumptions.

Misspecification of the parameters has a larger effect for MDSP samples. Under model 1 (table 22), misspecification of the correlation or variance does not have much effect, but misspecifying the mean can reduce the power of HE-COM-combination and SCORE3, making COMPOSITE2 the most powerful statistic. Under model 1' (table 23), misspecifying the variance or correlation hurts the power of HE-COM-combination and SCORE3 a bit, and misspecifying the mean cuts the power of those two statistics substantially. COMPOSITE2 is clearly the statistic with the most robust power for model 1'.

If one is adopting parameter estimates from a previous study it is likely that all three parameters will be incorrect by at least some margin. Since COMPOSITE2 and IBD2 do not use any parameter estimates, they should also be the most robust statistics for MDSPs and EDSPs respectively when there are errors in more than one parameter estimate. We did a limited study of the effects of misspecifying two parameters at a time. Detailed results are not shown, but the general qualitative result was that power was driven by how badly the mean was misspecified. This is consistent with the "one parameter wrong" runs described above, in which the mean had by far the greatest effect on power.

5.5. DISCUSSION

We have reviewed a number of new sibling pair QTL mapping statistics and investigated their appropriateness for discordant sibling pairs. For EDSPs, the best of the new statistics has only slightly higher power than the traditional IBD-sharing statistic, and the IBD-sharing statistic has the advantage of not depending on parameter estimates. For MDSPs, however, the statistics that

combine IBD-sharing information and correlation information (the group C statistics) substantially out-perform the IBD-sharing statistic. Of the group C statistics, COMPOSITE2 appears to be the most robust, having the highest power for non-normal models and being independent of trait parameter estimates.

Our results for EDSPs and MDSPs are interesting, but they are only a small piece of the story. The real importance of our results lies in two general conclusions that can be reached. The first is that any studies that use multiple-proband ascertainment would probably benefit from using group C-type statistics. The second is that the existence of the group C statistics makes it possible to explore entirely new experimental designs. When the only statistic for discordant pairs was IBD sharing, the range of designs was limited essentially to different definitions of extreme discordance. But with statistics that have robust ability to draw power from both the IBD sharing and the correlation between IBD and trait values, a much broader range of designs is possible. We have promoted the MDSP design as an option that might have the right balance of power and ease of ascertainment for some studies, but there are many other possibilities as well.

One way to think about new designs is to derive optimal designs for particular trait models. This approach was taken by Purcell et al. (2001), using the ascertainment-corrected variance components statistic of Sham et al. (2000). Given a trait model, they identified the most informative 5% of pairs and showed that the power of that ascertainment scheme was far higher than that of other ways of choosing 5% of pairs, even when the assumed trait model was wrong. Their method for identifying the most informative pairs could easily be applied to more moderate selection (e.g. selecting 15% or 30%) as well. The approach could be carried even further by assigning ascertainment cost/difficulty numbers to different pairs and selecting pairs to minimize total cost for a fixed amount of statistical power.

A very different way to think about new designs is to consider ascertainment schemes that are easy or convenient, and study their power. For example, a low-effort way to recruit discordant pairs might be to select extreme probands that are already enrolled in a clinical study and then recruit any siblings that are in, say, the opposite half of the trait distribution. This could lead to "discordant" pairs defined as one sibling in the top 10% of the distribution and one sibling in the bottom 50% of the distribution. With flexible statistics, such ascertainment need not even be precise. For example, in the hypothetical design just described, it is unlikely that a clinically-ascertained sample would actually be a random sample of the top 10% of the distribution; rather it might be made up of any individuals whose trait values were relatively high (without a uniform cutoff) and whose physicians referred them. Using a statistic like HE-COM-combination or SCORE3 frees us to consider designs with such imprecise ascertainment without having to worry about the validity of the statistical analysis. COMPOSITE2 is probably not a good choice when ascertainment is imprecise, because it depends on arbitrary weights. The equal weights we used performed very well for MDSPs as Forrest and Feingold (2000) defined them, but picking good weights for any other particular design would require some art and advance planning.

One type of "convenience sample" that deserves further study is affected sibling pairs already collected for linkage studies of binary traits. Affected sibling pairs can be considered concordant pairs for any quantitative traits associated with the disease they were originally collected to study (e.g. glucose and insulin levels for diabetes). Sibling pairs collected for linkage have been used to map QTLs, but it has been done with correlation-based statistics (e.g. Watanabe et al. 2000, Cai et al. 2001, Zhang et al. 2002). Such studies might in theory have much higher power with statistics like HE-COM-combination and SCORE3 that can also get information from the marginal IBD distribution. Again, we do not recommend COMPOSITE2 for

such non-traditional studies, because of the arbitrariness of the weights. It is also not clear whether the Haseman-Elston statistic is the right correlation-based statistic to form the composite with for anything other than discordant pairs. Huang and Jiang (2003) recently proposed a likelihood-based statistic for incorporating quantitative trait information into an affected sibling pair analysis, but it is not clear at this point how that method compares to the ones discussed here.

We did not explicitly study EDAC designs, but we believe that the same general results we have shown for discordant pairs will hold. We expect that the HE-COM-combination and SCORE3 statistics will have high and robust power. We suggest that variations on the EDAC design (for example choosing moderately discordant and concordant pairs) can be explored with those statistics.

We also recommend further study of designs that use larger sibships and even extended families that are selected based on two or more members with extreme phenotypes. As long as the selection is based on two or more people, the alternative-hypothesis IBD sharing is affected, and it is almost certainly beneficial to use a statistic that can capture that information. There is plenty of evidence that larger sibships are more powerful than sibling pairs, even in the context of discordant designs. Alcais and Abel (2000) and Tang and Siegmund (2001) both showed that if one has ascertained a discordant sibling pair, it is most efficient to also use any other siblings in the sibship - more efficient than recruiting another independent discordant pair. The score statistics extend in a natural way to larger sibships and extended pedigrees, and are probably the logical choice for studying and analyzing such designs. One problem that arises when a mix of pedigree types is used is that there may no longer be a good way to calculate an empirical variance of the IBD sharing in order to form a statistic that has the correct type I error. The statistic of Sham et al. (2002) attempts to deal with this problem.

In summary, we reiterate that the statistics we have investigated should open the door to a new era of studies using multiple-proband designs. The difficulty of recruiting EDSPs has for the most part kept such designs off the drawing table for the last few years, but it is now time for a renewed look at the possibilities.

Table 16 Genetic models – chapter 5

	Models						
	1	**2**	**3**	**4**	**5**	**1'**	**2'**
Type of inheritance	Add.	Dom.	Rec.	Add.	Dom.	Add.	Dom.
Locus heritability	0.2	0.2	0.2	0.2	0.2	N.A.	N.A.
Allele frequency	0.1	0.1	0.1	0.5	0.5	0.1	0.1
Trait means	-1, 0, 1	0, 1, 1	0, 0, 1	-1, 0, 1	0, 1, 1	-1.6, 0, 1.6	0, 1.6, 1.6
Env. std. dev.	0.849	0.785	0.199	1.414	0.866	N.A.	N.A.
Env. correlation	0.25	0.25	0.25	0.25	0.25	N.A.	N.A.
Overall mean	-0.8	0.19	0.01	0.0	0.75	-1.32	0.295
Overall std. dev.	0.949	0.877	0.222	1.581	0.968	2.047	1.393
Skewness	0.166	0.141	0.885	0.0971	-0.0991	-1.587	1.504
Kurtosis	0.0989	0.0235	3.814	0.0556	-0.0714	5.268	9.406
Overall Corr.	0.3	0.3	0.3	0.3	0.3	0.25	0.26

Table 17 Type I error (%) for EDSP samples

	Model													
	1		2		3		4		5		1’		2’	
	SD	Error	SD	Error	SD	Error	SD	Error	SD	Error	SD	Error	SD	Error
Group A														
IBD1	0.93	0.60	0.93	0.69	0.93	0.76	0.93	0.77	0.94	0.71	0.93	0.73	0.93	0.76
IBD2	1.00	0.98	1.00	0.95	1.00	1.06	1.00	1.14	1.01	0.99	1.00	0.97	1.01	1.05
Group B														
ORIGINAL.HE	1.00	0.96	1.01	1.13	1.02	1.22	1.00	1.03	1.02	1.05	1.01	1.06	1.01	0.82
TRAIT.SUM	1.00	0.95	1.01	0.99	1.02	1.06	1.01	0.91	1.00	0.97	1.02	1.13	1.01	0.84
TRAIT.PRODUCT	1.00	1.00	1.01	1.10	1.02	1.12	1.00	1.12	1.02	1.04	1.01	1.18	1.00	0.98
XU	1.00	1.10	1.00	0.98	0.95	0.80	1.01	1.09	1.01	1.10	0.98	0.89	0.98	0.88
V&H	0.75	0.15	0.75	0.17	0.39	0.00	0.76	0.15	0.78	0.13	0.49	0.01	0.54	0.02
FORREST	0.93	0.62	0.94	0.66	0.74	0.07	0.94	0.63	0.94	0.66	0.66	0.10	0.74	0.10
HE-COM-correlation	1.00	0.98	1.01	1.15	1.02	1.22	1.00	1.05	1.02	1.07	1.01	1.08	1.01	0.87
SCORE4	0.31	0.00	0.31	0.00	0.62	0.00	0.30	0.00	0.30	0.00	0.54	0.00	0.62	0.00
Group C														
HE-COM-combination	1.02	1.05	1.02	1.19	1.02	1.17	1.01	1.25	1.03	1.23	1.02	1.16	1.02	1.21
SCORE1	4.61	30.34	4.56	30.34	6.10	34.96	4.36	30.37	4.65	30.69	3.99	28.56	4.80	31.75
SCORE2	0.93	0.60	0.93	0.67	0.93	0.61	0.93	0.70	0.94	0.70	0.93	0.60	0.93	0.56
SCORE3	1.00	0.92	1.00	1.02	1.00	1.05	1.00	1.10	1.01	1.03	1.00	0.92	1.01	0.99
COMPOSITE1	0.96	0.76	0.97	0.90	0.97	0.83	0.96	0.91	0.98	0.96	0.97	0.75	0.97	0.73
COMPOSITE2	1.00	0.92	1.00	1.10	1.01	1.05	1.00	1.10	1.01	1.19	1.01	0.97	1.01	0.98

Note- the nominal type I error was set at 1%

Table 18 Power for EDSP samples at α=1% level

	Model						
	1	2	3	4	5	1’	2’
Group A							
IBD2	0.82	0.91	0.05	0.92	0.78	0.81	0.84
Group B							
ORIGINAL.HE	0.11	0.05	0.18	0.08	0.04	0.05	0.04
TRAIT.SUM	0.00	0.00	0.00	0.00	0.01	0.01	0.00
TRAIT.PRODUCT	0.11	0.05	0.17	0.08	0.05	0.07	0.04
XU	0.01	0.01	0.00	0.02	0.01	0.06	0.01
HE-COM-correlation	0.12	0.05	0.18	0.08	0.05	0.06	0.04
Group C							
HE-COM-combination	0.88	0.94	0.21	0.94	0.83	0.81	0.80
SCORE3	0.87	0.93	0.18	0.94	0.81	0.78	0.79
COMPOSITE2	0.79	0.78	0.22	0.81	0.62	0.66	0.71

Table 19 Power for MDSP samples at α=1% level

	Model						
	1	2	3	4	5	1’	2’
Group A							
IBD2	0.41	0.50	0.02	0.63	0.58	0.45	0.52
Group B							
ORIGINAL.HE	0.38	0.32	0.15	0.33	0.30	0.10	0.28
TRAIT.SUM	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TRAIT.PRODUCT	0.34	0.31	0.10	0.34	0.28	0.14	0.27
XU	0.03	0.03	0.00	0.04	0.02	0.10	0.06
HE-COM-correlation	0.37	0.32	0.15	0.34	0.29	0.11	0.30
Group C							
HE-COM-combination	0.73	0.77	0.13	0.84	0.79	0.43	0.64
SCORE3	0.72	0.77	0.12	0.84	0.79	0.42	0.64
COMPOSITE2	0.73	0.77	0.11	0.83	0.78	0.49	0.74

Table 20 Power for EDSP samples: sensitivity analyses under model 1

	$r = 0.1$	r = 0.5	$\mu = -1.75$	$\mu = 0.15$	$\sigma^2 = 0.45$	$\sigma^2 = 1.8$	*Population*
Group A							
IBD2	0.82	0.82	0.82	0.82	0.82	0.82	0.82
Group B							
ORIGINAL.HE	0.11	0.11	0.11	0.11	0.11	0.11	0.11
TRAIT.SUM	0.00	0.00	0.00	0.03	0.00	0.00	0.00
TRAIT.PRODUCT	0.11	0.11	0.04	0.13	0.11	0.11	0.11
XU	0.01	0.01	0.02	0.09	0.01	0.01	0.01
HE-COM-correlation	0.11	0.11	0.09	0.13	0.12	0.12	0.12
Group C							
HE-COM-combination	0.88	0.88	0.88	0.89	0.88	0.88	0.88
SCORE3	0.87	0.87	0.86	0.88	0.87	0.87	0.87
COMPOSITE2	0.79	0.79	0.79	0.79	0.79	0.79	0.79

Table 21 Power for EDSP samples: sensitivity analyses under model 1'

	$r = 0.1$	$r = 0.5$	$\mu = -3.37$	$\mu = 0.73$	$\sigma^2 = 2.10$	$\sigma^2 = 8.38$	*Population*
Group A							
IBD2	0.81	0.81	0.81	0.81	0.81	0.81	0.81
Group B							
ORIGINAL.HE	0.05	0.05	0.05	0.05	0.05	0.05	0.05
TRAIT.SUM	0.01	0.01	0.00	0.02	0.01	0.01	0.01
TRAIT.PRODUCT	0.07	0.07	0.04	0.08	0.07	0.07	0.07
XU	0.06	0.06	0.00	0.08	0.06	0.06	0.06
HE-COM-correlation	0.07	0.06	0.05	0.06	0.06	0.06	0.06
Group C							
HE-COM-combination	0.84	0.79	0.76	0.78	0.81	0.79	0.81
SCORE3	0.82	0.77	0.75	0.76	0.79	0.77	0.78
COMPOSITE2	0.66	0.66	0.66	0.66	0.66	0.66	0.66

Table 22 Power for MDSP samples: sensitivity analyses under model 1

	$r = 0.1$	r = 0.5	$\mu = -1.75$	$\mu = 0.15$	$\sigma^2 = 0.45$	$\sigma^2 = 1.8$	*Population*
Group A							
IBD2	0.41	0.41	0.41	0.41	0.41	0.41	0.41
Group B							
ORIGINAL.HE	0.38	0.38	0.38	0.38	0.38	0.38	0.38
TRAIT.SUM	0.00	0.00	0.00	0.06	0.00	0.00	0.00
TRAIT.PRODUCT	0.34	0.34	0.03	0.39	0.34	0.34	0.34
XU	0.03	0.03	0.05	0.38	0.03	0.03	0.03
HE-COM-correlation	0.36	0.38	0.26	0.45	0.37	0.37	0.37
Group C							
HE-COM-combination	0.71	0.73	0.60	0.75	0.72	0.73	0.73
SCORE3	0.71	0.73	0.59	0.74	0.72	0.72	0.72
COMPOSITE2	0.73	0.73	0.73	0.73	0.73	0.73	0.73

Table 23 Power for MDSP samples: sensitivity analyses under model 1'

	$r = 0.1$	$r = 0.5$	$\mu = -3.37$	$\mu = 0.73$	$\sigma^2 = 2.10$	$\sigma^2 = 8.38$	*Population*
Group A							
IBD2	0.45	0.45	0.45	0.45	0.45	0.45	0.45
Group B							
ORIGINAL.HE	0.10	0.10	0.10	0.10	0.10	0.10	0.10
TRAIT.SUM	0.00	0.00	0.00	0.00	0.00	0.00	0.00
TRAIT.PRODUCT	0.14	0.14	0.05	0.20	0.14	0.14	0.14
XU	0.10	0.10	0.01	0.34	0.10	0.10	0.10
HE-COM-correlation	0.12	0.10	0.07	0.21	0.11	0.11	0.11
Group C							
HE-COM-combination	0.49	0.40	0.17	0.38	0.45	0.37	0.43
SCORE3	0.48	0.40	0.17	0.37	0.44	0.36	0.42
COMPOSITE2	0.49	0.49	0.49	0.49	0.49	0.49	0.49

6. A NEW LINKAGE STATISTIC FOR DISCORDANT SIBLING PAIRS OUTPERFORMS CURRENT STATISTICS.

6.1. SUMMARY

Chapter 5 evaluated the performance of a wide variety of statistics for QTL linkage using discordant sibling pairs. They found that the most powerful statistics in general were a score statistic and a "composite statistic." However, they pointed out that while these two statistics have equal power under ideal conditions, each has limitations that reduces its power in certain circumstances. The score statistic depends on estimates of trait parameters, and can lose a great deal of power if those estimates are incorrect. The composite statistic is not sensitive to trait parameter estimates, but does depend on arbitrary weights that must be chosen based on the ascertainment scheme. In this report we elucidate the algebraic relationship between the score and composite statistics, and then use that relationship to suggest a new statistic that should combine the best properties of both. We call our new statistic the "robust discordant pair" (RDP) statistic. We report on simulation studies showing that the RDP statistic does indeed have all of the strengths and none of the weaknesses of the score and composite statistics.

6.2. METHODS

Chapter 5 used simulation studies to evaluate the type I error and power of various statistics for QTL linkage using discordant sibling pairs. They considered a number of statistics from the

literature, as well as several new variants. The bottom-line result was that three statistics looked best, by virtue of having correct type I error across the board and power higher than other statistics for all or most trait models tested. All three of the best statistics are of the "combination" type - that is, they draw linkage information from the marginal IBD-sharing distribution and also from the correlation between IBD sharing and trait values.

One of the three best statistics is a score statistic. Tang and Siegmund (2001) derived the basic statistic, and Szatkiewicz et al. (2003) (e.g. chapter 5) proposed a variant (SCORE3 in their terminology) that has an entirely empirical variance in the denominator. The formula for the score statistic with the empirical variance estimate is

$$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - 1/2)^2\right]}},$$

where π_i is the estimated IBD-sharing proportion for pair i, and

$$A_i = \frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2} + \frac{4r}{1-r^2}.$$

The parameter r is the sibling correlation for the trait. Y_{iS} and Y_{iD} are the squared trait sum and the squared trait difference, respectively, calculated based on trait values that are standardized to have mean zero and variance one. That is, if X_{i1} and X_{i2} are the trait values for pair i, then

$$Y_{iS} = \left[\frac{(X_{i1}-\mu)}{\sigma} + \frac{(X_{i2}-\mu)}{\sigma}\right]^2 \text{ and } Y_{iD} = \left[\frac{(X_{i1}-\mu)}{\sigma} - \frac{(X_{i2}-\mu)}{\sigma}\right]^2,$$

where μ is the population trait mean and σ is the population trait standard deviation. Thus the formula for A_i effectively involves three population trait parameters: the mean, the variance, and the sibling correlation. Chapter 5 showed that with good estimates of these three parameters, this statistic has just about the maximal possible power for reasonably Gaussian trait models. However, if the parameters (particularly the mean) are not estimated well, the score statistic can lose a substantial amount of power.

The second good statistic for discordant pairs according to Chapter 5 is almost identical to the score statistic. It is one version of the HE-COM statistic proposed by Sham and Purcell (2001). Sham and Purcell proposed two different statistics under the name "HE-COM." The first is *not* a combination statistic (it uses only the information on the correlation between IBD-sharing and trait values), but the second *is* a combination statistic. Szatkiewicz et al. (2003) referred to these two statistics as S&P1 and S&P2. Here we use the more informative names "HE-COM-correlation" and "HE-COM-combination." HE-COM-combination is identical to the score statistic described above, except that it uses a slightly different variance estimate in the denominator.

The third statistic that performed well in the studies by chapter 5 is a variant of the "composite statistic" originally proposed by Forrest and Feingold (2000). The composite statistic is a weighted average of the Haseman-Elston regression statistic (Haseman and Elston 1972) and an identity-by-descent (IBD) sharing statistic. The Haseman-Elston statistic is based on the squared trait difference, $(X_{i1} - X_{i2})^2$, which does not involve any population trait parameters. Because the composite statistic does not depend on trait parameters, it avoids the parameter misspecification problems of the score statistic. However, in order for the composite statistic to have power as high as that of the score statistic, the weights for the two components must be

intelligently chosen, using some knowledge of the ascertainment scheme. Forrest and Feingold (2000) showed that it is not too difficult to find good weights if the ascertainment scheme is known, but it would be better if the weighting issue could be eliminated completely.

We show here that the score statistic can be algebraically decomposed into a Haseman-Elston type statistic plus an IBD-sharing statistic, with weights that are data-dependent. This helps to explain the relationship between the score and composite statistics, and also suggests a statistic that combines the best properties of both. The decomposition is as follows.

$$\text{score statistic} = \frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - 1/2)^2\right]}}$$

$$= \frac{\sum A_i(\pi_i - \bar{\pi}) + \sum A_i(\bar{\pi} - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - 1/2)^2\right]}}$$

$$= \frac{\sum A_i(\pi_i - \bar{\pi})}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum(\pi_i - 1/2)^2\right]}} + \left(\frac{\sum A_i}{\sqrt{n\sum A_i^2}}\right)\frac{\bar{\pi} - 1/2}{\sqrt{\left[\frac{1}{n^2}\sum(\pi_i - 1/2)^2\right]}}.$$

The first term of the decomposed score statistic is essentially HE-COM-correlation, with an empirical variance in the denominator rather than a regression-based variance estimate. The second term is an IBD-sharing statistic, again with an empirical variance estimate in the denominator. The two components are weighted by a factor that is large when the A_i's have large

absolute value (as in discordant pair samples) and is zero (expected value) when the A_i's have mean zero (as in population samples).

Viewed through the lens of this decomposition, there are only two differences between the score statistic and the composite statistic. The first difference is that the composite statistic uses arbitrary fixed weights for the two components, while the score statistic uses data-dependent weights that automatically adapt to the sampling scheme. For example, if both statistics were applied to a population sample, the composite statistic would perform quite poorly because it would be incorporate random noise from the IBD-sharing statistic, while the score statistic would automatically adjust the weight on the IBD-sharing statistic to zero and thus still have maximal power. If both statistics were applied to a sample of discordant pairs, the score statistic would again automatically adjust the weight to be optimal no matter what the ascertainment rule, whereas the composite statistic would only perform well if we had pre-chosen a good weight. The second difference between the score statistic and the composite statistic is that the score statistic is based on the trait function A_i, which involves the trait difference, the trait sum, and the three population trait parameters. The composite statistic by contrast is based on the original Haseman-Elston regression procedure and uses the trait function $(X_{i1} - X_{i2})^2$, which does not invlove the trait sum or the trait parameters. For most types of samples the function A_i contains more information than the squared trait difference alone (Sham and Purcell 2001), but our previous simulation results suggest that for discordant pairs the information is almost equal (Forrest and Feingold 2000, chapter 5).

If we combine the data-dependent weights of the score statistic with the parameter-independent trait function of the composite statistic, we ought to get a new statistic with the best

features of both. We propose the following statistic, which we term the robust discordant pair (RDP) statistic:

$$\frac{\sum (X_{i1} - X_{i2})^2 (\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left[\sum (X_{i1} - X_{i2})^4\right]\left[\sum (\pi_i - 1/2)^2\right]}},$$

which is exactly the same as the score statistic, but with $(X_{i1} - X_{i2})^2$ substituted for A_i.

6.3. RESULTS

We updated the simulation studies in chapter 5 to include the RDP statistic. The methods are described in detail in chapter 5. Table 24 below reports power (based on 1000 replicates) for selected statistics applied to extreme discordant sibling pairs (EDSPs), defined as one sibling in the top 10% of the trait distribution and the other sibling in the bottom 10%. Table 25 shows the analagous results for moderately discordant sibling pairs (MDSPs), defined as one sibling in the top 35% of the trait distribution and the other sibling in the bottom 35%. Trait models 1 - 5 are simple mixture-of-normals models (see chapter 5 for model details), and models 1' and 2' are non-Gaussian models that were created by transforming models 1 and 2. Model 3 is a recessive model that has substantial skewness and kurtosis. All statistics shown have correct type I error.

Several important features of the statistics are evident from table 24. First, the score statistic with correct parameter estimates, the composite statistic with extreme weights, and the RDP statistic all have essentially equal power for the reasonably Gaussian models (1, 2, 4, and 5). These three statistics are marked by bold typeface in the table. The composite statistic with

equal weights has much lower power, underscoring the importance of the weights. For EDSPs, the score statistic does *not* lose much power if the mean is misspecified, because for extreme pairs most of the linkage information comes from the IBD-sharing. Table 25, for MDSPs, shows essentially the same features of the statistics, except that here the score statistic *does* lose a substantial amount of power when the mean is misspecified. We have not shown results for misspecification of the other parameters. The effect of misspecifying other parameters follows similar patterns, but is smaller overall (see chapter 5 for partial results).

The performance of the statistics for the non-Gaussian models requires separate comment. The score statistic is derived under a normality assumption, so it should not necessarily perform well for non-Gaussian models. This is also somewhat true for the RDP statistic, whose form follows from that of the score statistic. For models 1' and 2', the composite statistic, which does not depend on the normality assumption, does have higher power than the score statistic and the RDP statistic. This is not true, however, for model 3, which is also substantially skewed. We feel that the behavior of the statistics for non-Gaussian models still requires further study. It is not clear what types of non-Gaussian trait models are most realistic and important, and it is also not clear how various features of the models and statistics interact to determine which statistic is most powerful.

6.4. CONCLUSION AND DISCUSSION

Overall, we recommend our new RDP statistic as the best choice for discordant sibling pair studies in just about any situation. It has power equal to that of the score and composite statistics, but is robust to parameter misspecification and does not depend on arbitrary weights. For EDSPs it is probably fine to use the score statistic or even the IBD-sharing statistic instead, but using the

RDP statistic adds an extra measure of robustness at no cost in power. For more moderately-selected samples, the RDP statistic is clearly preferable. Further study is required before we can make recommendations about statistics for substantially non-Gaussian trait models, but based on our results to date the RDP statistic and the composite statistic (with appropriate weights) seem to be the best choices.

.

Table 24 Power (%) for extreme discordant sibling pairs (EDSP) at α=1% level

	Models						
	1	2	3	4	5	1'	2'
score statistic - correct parameters	**87**	**93**	**18**	**94**	**81**	**78**	**79**
score statistic - mean estimate low by one SD	86	93	18	93	80	75	74
score statistic - mean estimate high by one SD	88	94	21	93	80	76	77
composite statistic - equal weights*	79	78	22	81	62	66	71
composite statistic - extreme weights**	**86**	**94**	**12**	**94**	**82**	**84**	**87**
robust discordant pair statistic	**87**	**93**	**19**	**93**	**81**	**78**	**79**

Note - * Forrest and Feingold (2000) recommended equal weights for MDSPs,
** Forrest and Feingold (2000) recommended the weights (.259, .966) for EDSPs.

Table 25 Power (%) for moderately discordant sibling pairs (MDSP) at α=1% level

	Models						
	1	2	3	4	5	1'	2'
score statistic - correct parameters	**72**	**77**	**12**	**84**	**79**	**42**	**64**
score statistic - mean estimate low by one SD	59	65	12	79	77	17	28
score statistic - mean estimate high by one SD	74	77	15	78	67	37	42
composite statistic - equal weights*	**73**	**77**	**11**	**83**	**78**	**49**	**74**
composite statistic - extreme weights**	58	67	4	79	72	53	66
robust discordant pair statistic	**71**	**76**	**12**	**84**	**78**	**42**	**67**

Note - * Forrest and Feingold (2000) recommended equal weights for MDSPs.
** Forrest and Feingold (2000) recommended the weights (.259, .966) for EDSPs.

7. QTL MAPPING WITH DISCORDANT AND CONCORDANT SIBLING PAIRS - NEW STATISTICS AND NEW DESIGN STRATEGIES.

7.1. SUMMARY

Gu et al. (1996) proposed the so-called "extreme discordant and concordant" (EDAC) sampling design for quantitative trait locus mapping using sibling pairs. The principle of the design is to gain efficiency by genotyping only the most informative of the available sibling pairs. EDAC-type designs have been studied and extended in a number of papers, and have been applied in a few others. This literature is somewhat out of date, however, in that there are many new statistics that are appropriate for EDAC data. With newer statistics, the power of EDAC designs can be improved. Moreover, the relative power of different designs must be re-evaluated, because the newer statistics improve the power of some designs more than others. In this chapter we review a number of available design and statistic choices for EDAC studies, and use simulation to show what statistics are most powerful for each design. We then use those more powerful statistics to suggest strategies for making design choices among various EDAC and non-EDAC designs that use sibling pairs. We find that when genotyping must be minimized, an EDAC design with predominantly discordant pairs is the best choice, and when a balance of genotyping and phenotyping effort must be achieved, single proband ascertainment can do better. We also show that moderately selected samples (as opposed to very extreme samples) can be an efficient choice for many studies.

7.2. INTRODUCTION

It has been amply demonstrated that phenotypically selected sibling pairs are usually more powerful than population samples of sibling pairs for linkage analysis of a quantitative trait locus (QTL) (e.g., Carey and Williamson 1991; Eaves and Meyer 1994; Risch and Zhang 1995). Moreover, it is clear that under most genetic models the majority of the power to detect linkage is concentrated in three types of sibling pairs: those in which both siblings have high trait values (high concordant pairs), those in which both siblings have low trait values (low concordant pairs), and those in which one sibling has a high trait value and the other has a low trait value (discordant pairs) (e.g. Risch and Zhang 1995, Purcell et al. 2001). One of the best-known strategies for taking advantage of this information is the extreme discordant sibling pair (EDSP) design (Risch and Zhang 1995), in which one samples discordant pairs with one sibling in the top 10% of the trait distribution and the other in the bottom 10%. Gu et al. (1996) pointed out that any method designed to select a certain number of extreme discordant sibling pairs would necessarily identify many concordant pairs during the screening process. They proposed combining extreme discordant and concordant (EDAC) siblings, and argued that this design is more likely to maximize power and to be cost effective than study designs pursuing only extreme discordant sibs.

A number of variations on EDAC designs have since appeared in the literature (e.g., Gu and Rao 1997, Dolan and Boomsma 1998, Purcell et al. 2001). For example, the concordant pair portion of the sample can include either high concordant pairs, low concordant pairs, or both. One can select both concordant and discordant pairs according to the same percentile threshold (e.g. top and bottom 10%), or use different thresholds for concordant and discordant pairs in order to achieve a particular ratio of discordants to concordants. Or, one can abandon thresholds

entirely in favor of a scheme that chooses the most informative pairs under particular assumptions (Purcell et al. 2001, Sham and Purcell 2001).

In addition to the variety of designs, quite a variety of test statistics has been used. The statistic originally proposed by Gu et al. (1996) is an equally-weighted average of the IBD-sharing statistics for the discordant and concordant samples. Li and Zhang (2000) proposed a similar test, and Li and Gastwirth (2001) extended it to incorporate weights that depend on the extremity of the sib pair. In an EDAC study of blood pressure, Xu et al. (1999) analyzed concordant high, concordant low and discordant pairs individually using an IBD-sharing statistic for each. Purcell et al. (2001) explored optimal sampling strategies using a conditional variance-components statistic, and Sham and Purcell (2001) used their HE-COM method. Fullerton et al. (2003) used IBD-sharing tests on discordant and concordant pairs separately, and also combined all pairs in a regression analysis using the method proposed by Visscher and Hopper (2001). Starting from probands with extremely low bone mineral density, Wilson et al. (2003) recruited EDAC sib pairs and any available additional sibs from 254 pedigrees and performed nonparametric multipoint linkage analysis (MAPMAKER/SIBS v.2.0) on all possible pairs. Most recently, van Asselt et al. (2004) analyzed EDAC data from 130 families by an adjusted variance-components method (Morton 1959) that requires clearly defined selection thresholds. In addition, they applied the Haseman-Elston (1972) regression approach to subgroups and to a slightly larger dataset, all of which included sibs outside of the strict selection region.

The current state of affairs is that investigators considering EDAC studies will find many unanswered questions about both designs and test statistics. The statistics mentioned above are only a small subset of those that might be used; chapter 5 evaluated a wide variety of statistics for EDSP designs, and many of the statistics they considered should be appropriate and powerful

for EDAC samples as well. It is also not entirely clear what design choices are best under what circumstances. A number of papers have compared different EDAC designs or compared EDAC to non-EDAC designs (e.g. Gu et al. 1996, Gu and Rao 1997), but those comparisons are limited because they rely on older statistics. The most informative comparison should use the best statistic for each design, and thus these design comparisons need to be updated now that new statistics are available.

In this study we attempt to fill these gaps and bring the literature on EDAC designs up to date. We first review the most important design issues and the available test statistics. We then use simulation studies to show what statistics are most powerful for each design. Finally, we use these most powerful statistics to compare designs and make suggestions about what sampling strategies are best under what circumstances.

7.3. EDAC DESIGN CHOICES

In our view there are four decisions that are most critical in choosing an EDAC design. They are 1) threshold-based sampling vs. "optimal" sampling, 2) both high and low concordants vs. just one or the other, 3) ratio of discordant to concordant pairs, and 4) extreme vs. moderate sampling. We discuss each of these in turn below.

7.3.1. Threshold-based sampling or optimal sampling?

The classical EDAC design chooses sibling pairs according to fixed thresholds, typically percentiles of the trait distribution. For example, panel A of figure 4 on page 149 shows a sample chosen such that discordant pairs have one sibling in the top 12% and one in the bottom 12%, while concordant pairs must both be in the top or bottom 4%. An alternative selection strategy, proposed by Purcell et al. (2001) and Sham and Purcell (2001), is to calculate the

informativeness (non-centrality parameter) of each phenotyped sibling pair and choose precisely the optimal subset to genotype. Panel F of figure 4 on page 149 shows a sample drawn according to the method outlined in Sham and Purcell (2001).

While "optimal" sampling would seem by definition to be better than threshold-based sampling, there are some complications. One is that calculating the informativeness of a given sib pair depends on an assumed trait model. However, Purcell et al. (2001) proposed using a "base model" with an additively-acting allele of 50% frequency, and showed that selection based on their base model is much more powerful than threshold-based selection even when the base model is incorrect. The more important complication is that the optimality of "optimal" sampling is based only on genotyping. If genotyping cost is not the only consideration, then optimal sampling may not be the best choice. Optimal sampling requires recruiting and phenotyping a whole sample of sibling pairs and then deciding which pairs to genotype. With threshold-based sampling, recruitment can be rolling, and if the first sibling phenotyped is not above or below the threshold, the second sibling need not be recruited at all, let alone phenotyped. In our later section entitled Comparison of Sampling Designs, we give a more quantitative basis for comparing optimal and threshold-based sampling.

7.3.2. High concordant pairs, low concordant pairs, or both?

Numerous authors have demonstrated that the linkage information shifts from one type of extreme concordant pair to the other as the allele frequency and the degree of dominance vary (e.g. Risch and Zhang 1995, Gu et al. 1996, Allison et al. 1998). Basically, the tail associated with the rarer genotype is more informative. While Gu et al. (1996) proposed including both high and low concordant pairs (e.g. panels A and B of figure 4, which we refer to to as four-corner sampling), Gu and Rao (1997) explored variants that included only one or the other (e.g. panels

C and D of figure 4,which we refer to as three-corner sampling). One way of interpreting their findings is to suggest that one should use only high concordant pairs or only low concordant pairs given enough knowledge of the trait model, but that if the trait model is unknown it is preferable to use both low and high concordants to keep the power robust.

We prefer a slightly different way of looking at the question, which is to assume that we are studying a complex trait with many different genes influencing it. Some genes will affect mostly one tail or the other, and some will affect both. Then the decision the researcher has to make is really whether he/she is *interested in* both tails or only one. If one tail is of primary interest (as might be the case in a blood pressure or cholesterol study for example) then only concordants from that tail should be included in EDAC design. If variation over the entire spectrum is of interest, then both tails should be sampled. An interesting example is the subgroups analyses for early and late menopausal age conducted by van Asselt et al. (2004). In that study they ascertained a four-corner sample, but also performed analyses on three-corner subsamples.

7.3.3. Ratio of discordant to concordant pairs

Another important design issue in EDAC studies is what the best combination of extreme discordant and extreme concordant pairs is. This issue is only applicable to threshold-based sampling; for optimal sampling it is decided automatically.

Under most trait models, discordant pairs contain more linkage information than concordant pairs. In particular, when there is a positive sibling correlation due to environmental and/or polygenic components, the power of concordant pairs decreases (Risch and Zhang 1995). It was noted (Gu et al. 1996, Gu and Rao 1997) that including all available extreme concordant pairs with discordant pairs may reduce power per genotype, and that the optimal combinations of

extreme concordant and discordant pairs are dependent on the trait model. Dolan and Boomsma (1998) conducted an extensive investigation of whether such optimal selection in EDAC designs can be achieved when prior knowledge of the genetic model is limited or absent. Based on a power formula for the Gu et al. (1996) EDAC test, they calculated and then averaged the best selection percentages over a great number of different genetic models, and they recommended allowing the selection percentage of extreme discordant pairs to exceed the percentage of extreme concordant pairs by a factor of 1.9 to 2.6 (Table I of Dolan and Boomsma 1998.) Purcell et al. (2001) also found that discordant pairs generally outnumbered concordant pairs in their optimal samples. However, most applied studies to date have used the same percentile threshold for both discordant and concordant pairs, and thus have ascertained far more concordants than discordants.

In our Comparison of Sampling Designs section we look at whether a preponderance of discordant pairs is still the preferred EDAC design when new statistics are used. We also compare phenotyping and genotyping numbers for designs with a preponderance of discordant pairs to those for designs that use the same threshold for both types of pairs.

7.3.4. Extreme or moderate sampling?

For either threshold-based or optimal sampling, one must decide how "deep" to sample, i.e., how extreme the discordant and concordant pairs should be. Under most circumstances, the more extreme the pair the more linkage information it carries (Risch and Zhang 1995). Traditional EDAC and EDSP designs have relied on very extreme pairs and, consequently, they require screening enormous numbers of pairs. That is probably the reason that these designs are only occasionally used in practice. For example, Fullerton et al. (2003) screened 34,580 sibling pairs in order to choose 408 discordant pairs, 414 low-concordant pairs, and 410 high-concordant

pairs. Several authors (Gu and Rao 1997, Forrest and Feingold 2000) have proposed using less restrictive thresholds and sampling "moderately" discordant pairs in order to achieve a balance between screening and genotyping efforts. For EDSP designs, Forrest and Feingold (2000) and chapter 5 demonstrated that, with proper statistics, moderately discordant sibling pairs can be powerful and practical. For EDAC designs, extreme and moderate sampling strategies have been compared using the original Gu et al. (1996) IBD-sharing statistic (Gu and Rao1997, Dolan and Boomsma 1998), but since that statistic is far from optimal for moderate selection (Forrest and Feingold 2000), this comparison needs to be updated. In our Comparison of Sampling Designs section we give results that can be used to decide how extreme a sample is desirable for a particular study.

7.4. EDAC TEST STATISTIC CHOICES

Whenever a sibling pair is selected based on the phenotypes of both individuals, the IBD-sharing distribution is potentially distorted at any marker linked to the trait. If the selection is extreme, the change in the IBD sharing can be extreme. This is the rationale behind the combination of extreme selection and IBD-sharing statistics that has traditionally been used in EDSP and EDAC studies. By contrast, most human QTL-mapping statistics (e.g. maximum likelihood variance components, most regression-based methods) do not consider the marginal distribution of IBD sharing at all. Instead, they base their statistical power on looking for correlation between each pair's IBD sharing and some measure of the similarity of trait values. We describe these methods as "correlation-based" methods; examples include maximum-likelihood variance components, the regression method of Haseman and Elston (1972), and many of the newer regression-based methods (see Feingold 2002 for a more detailed discussion). IBD-sharing statistics are powerful

for extreme samples, but are far from optimal for more moderately selected samples (Forrest and Feingold 2000, chapter 5). But correlation-based statistics are not ideal for moderately selected discordant or concordant pairs either, because they do not capture the linkage information in the IBD sharing. If we want to consider a rich variety of sampling designs and get maximal power from each design, we need to consider statistics that combine both the correlation information and the IBD-sharing information.

Forrest and Feingold (2000) made the critical observation that IBD-sharing statistics and correlation-based statistics are independent under both the null and alternative hypotheses, and so may be combined to give a more powerful test of linkage. The original proof of independence offered by Forrest and Feingold (2000) is overly complex and contains an error, but the result is correct. In the appendix A to this dissertation we offer a simpler proof of independence for the statistics we consider here. Several statistics have appeared in the literature that combine both the correlation information and the marginal IBD-sharing information, and we term these "combination" statistics. (See chapter 5 for a more thorough review of these types of statistics.) One way to look at the relationship among the IBD-sharing statistics, the correlation-based statistics, and the combination statistics is to note that the computational formulas for the statistics can be simply partitioned. Let π_i be the estimated IBD sharing for pair i, $\bar{\pi}$ the average IBD-sharing over all the pairs, and let A_i be some function of the sib's trait values (e.g. the squared trait difference for Haseman-Elston regression). The IBD-sharing statistic for a set of pairs is generally $\bar{\pi} - 1/2$, standardized by some appropriate variance estimate. The correlation-based statistics are generally of the form $\sum A_i(\pi_i - \bar{\pi})$, again standardized by some appropriate variance estimate, which is independent of $\bar{\pi} - 1/2$. Most of the combination statistics, by contrast, are of the form $\sum A_i(\pi_i - 1/2)$ (again, standardized). But $\sum A_i(\pi_i - 1/2)$ can be written as $\sum A_i(\pi_i - \bar{\pi}) +$

$\left(\bar{\pi} - 1/2\right)\sum A_i$, showing that the combination statistics are weighted sums of correlation-based statistics and IBD-sharing statistics. Chapter 6 used this decomposition for discordant sib pairs to suggest a new combination statistic that outperforms all existing statistics.

In this chapter we have tried to limit attention to the statistics that are most likely to be important for EDAC samples. We conducted our simulation studies with all of the statistics considered by chapter 5 and chapter 6 for EDSP designs, as well as several EDAC-specific statistics. However, we report results in this chapter only for the statistics that are particularly useful for EDAC data. We do not even mention, for example, any of the statistics reviewed by chapter 5 that did not have correct type I error for selected samples.

7.4.1. IBD-sharing statistic

We report power results for only one IBD-sharing statistic, the Gu et al. (1996) statistic with an empirical variance estimate, designated as "Gu-empirical.variance" hereafter. The formula of this statistic is

$$\frac{\bar{\pi}_{CS} - \bar{\pi}_{DS}}{\sqrt{\hat{\sigma}^2_{pooled}\left[\frac{n_{CS} + n_{DS}}{n_{CS} \cdot n_{DS}}\right]}},$$

where n_{CS} is the total number of concordant pairs, n_{DS} is the number of discordant pairs, $\bar{\pi}_{CS}$, $\bar{\pi}_{DS}$ are the average estimated IBD sharing among concordant pairs and discordant pairs respectively, and $\hat{\sigma}^2_{pooled}$ is the pooled empirical variance among all the sib pairs assuming homoscedasticity under the null hypothesis (i.e. $\sigma^2_{CS} = \sigma^2_{DS}$). In the original EDAC test proposed by Gu et al. (1996), the variance in the denominator is a theoretical value that assumes IBD information for each pair is observed perfectly (i.e. that the marker is infinitely polymorphic). We use the empirical variance estimate rather than the theoretical one so that the statistic will have

correct type I error even when the marker is not completely informative (see chapter 5 for further discussion).

7.4.2. Correlation-based statistics

We report results for two example correlation-based statistics, the original Haseman-Elston (1972) regression statistic and the HE-COM statistic (Sham and Purcell 2001). Note that Sham and Purcell (2001) actually discussed two different statistics under the name HE-COM: one is a correlation-based statistic and the other is a combination statistic. Szatkiewicz et al. (2003) referred to these as S&P1 and S&P2. Here we use the more informative names HE-COM-correlation and HE-COM-combination. Sham and Purcell (2001) confirmed the equivalence between the HE-COM-correlation and variance components (e.g. Amos 1994, Almasy and Blangero 1998) methods for population samples, and they recommended HE-COM-combination for any selected sample.

7.4.3. Combination statistics

We consider four combination statistics. The first is a score statistic that was designated as "SCORE3" in Szatkiewicz et al. (2003) (e.g. chapter 5). It has the formula

$$\frac{\sum A_i(\pi_i - 1/2)}{\sqrt{\frac{1}{n}\left(\sum A_i^2\right)\left[\sum (\pi_i - 1/2)^2\right]}}.$$

As previously defined, π_i is the estimated mean IBD sharing for sibling pair i. A_i is a function of the pair's trait values and model parameters and is defined as

$$A_i = \frac{Y_{iS}}{(1+r)^2} - \frac{Y_{iD}}{(1-r)^2} + \frac{4r}{1-r^2},$$

$$Y_{iS} = \left[\frac{(x_{i1}-\mu)}{\sigma} + \frac{(x_{i2}-\mu)}{\sigma}\right]^2,\ Y_{iD} = \left[\frac{(x_{i1}-\mu)}{\sigma} - \frac{(x_{i2}-\mu)}{\sigma}\right]^2,$$

where x_{i1}, x_{i2} are the actual trait values for sibling pair *i*, μ and σ are the population trait mean and standard deviation, and *r* is the population sibling pair correlation.

The second combination statistic is the "robust discordant pair" (RDP) statistic proposed by chapter 6. The RDP statistic is exactly the same as SCORE3, but with $(x_{i1} - x_{i2})^2$ substituted for A_i. Consequently, it is parameter free and it was shown to be the best statistic for discordant sib pairs among all existing methods (chapter 6). We do not expect it to perform well for EDAC samples; it is included here because it serves as a component in our "RDP composite" statistic below.

Note that both the RDP statistic and SCORE3 have empirical denominators that give consistently correct type I error across trait models and sampling schemes. SCORE3 is derived under a normality assumption, so it should not necessarily perform optimally for non-normal models, and we anticipate the same property at least to some extent for the RDP statistic.

The third and fourth combination statistics we consider are both composite statistics of the type originally described by Forrest and Feingold (2000). That is, they are weighted sums of independent statistics. Forrest and Feingold (2000) used a weighted sum of the Haseman-Elston (1972) statistic and the IBD-sharing statistic for discordant pairs. For EDAC samples, we propose two variants. Our "RDP composite" statistic is constructed as

$$w_1 RDP_{discordant} + w_2 IBD_{concordant},$$

where $RDP_{discordant}$ is the RDP statistic applied to the discordant pairs in the sample, and $IBD_{concordant}$ is the mean IBD-sharing statistic applied to the concordant pairs in the sample. Our "3-part composite" statistic further decomposes the $RDP_{discorant}$ and constructs a composite statistic as a weighted sum of the IBD-sharing statistic for the discordant pairs, the IBD-sharing statistic for the concordant pairs, and the Haseman-Elston regression statistic for the discordant pairs only:

$$w_1 HE_{discordant} + w_2 IBD_{discordant} + w_3 IBD_{concordant}.$$

Note that the IBD-sharing statistic appearing in both composite statistics should be computed with an empirical variance as:

$$\frac{\overline{\pi} - 1/2}{\sqrt{\sum (\pi_i - 1/2)^2 / n^2}}.$$

Based on extensive testing, we concluded that a component measuring correlation for the concordant pairs should not be included in either of the composite statistics because it adds more noise than signal in most cases. This is equivalent to saying that there is not much information in the actual trait values among the concordant pairs. It was somewhat surprising to us that this was true even for moderately ascertained samples.

For each of the composite statistics, we followed the procedure outlined in Forrest and Feingold (2000) to determine sensible weights. First we conducted small simulation studies to

find weights that yield maximum power for different EDAC samples under a variety of models. Consistent with the results of Forrest and Feingold (2000), we found that the weights depend primarily on the ascertainment scheme, not on the trait model. We then simply implemented model 1's weights for each of the four ascertainment schemes. Those weights reflect the relative strength of each component and also adjust the signs so that the absolute values of each component are summed.

7.5. COMPARISON OF TEST STATISTICS

We performed simulation studies to find the best test statistic for each type of EDAC design. We report here results for moderate and extreme three-corner and four-corner samples. All of these samples use unequal thresholds and have a preponderance of discordant pairs. We report separately in the Comparison of Sampling Designs section on studies using equal thresholds. We did not perform simulation studies with optimal samples, but it is fairly clear from our results that the same statistics are most powerful for almost all sensible EDAC variations including optimal sampling.

7.5.1. Simulation methods

Table 26 lists the eleven trait models considered in this study, which were also described previously in chapter 4 and chapter 5. Briefly, we model a diallelic QTL which has the increaser allele with frequency p ranging from 0.1 to 0.9, and gene action additive, dominant, or recessive. In each of models one through nine, we added a normally distributed environmental effect to the genotype scores for each sibling, with the two siblings having a correlation of .25 to account for polygenic and common environmental components. The means and variances were chosen to give each model a locus heritability of .2. Note the symmetry between certain pairs of models: 1

and 7; 2 and 9; 3 and 8; and 5 and 6. Furthermore, we considered two hypothetical non-normal models by applying an arbitrary transformation procedure. That is we generated models 1' and 2' by simulating data under models 1 and 2, respectively, and then taking the signed square ($x|x|$) of each trait value, which yields overall trait distributions that are somewhat skewed and have high kurtosis.

Under each of the trait models, we simulated nuclear families with two children and ascertained siblings using four different methods. The first is extreme three-corner sampling (EDAC-3corner), which selects high concordant sib pairs that are both in the top 4% of the trait distribution and discordant sib pairs with one sibling is in the top 12% of the trait distribution and the other sibling in the bottom 12%. The second selection scheme is extreme four-corner sampling (EDAC-4corner), which simply adds the low concordant pairs (at the 4% threshold). The third and fourth schemes are moderate three- and four-corner samples (MDAC-3corner and MDAC-4corner), which use 24% and 8% thresholds instead of 12% and 4%. Panels A through D of figure 4 on page 149 illustrate samples under each of these four schemes. Our thresholds of 12% and 4% were based on the recommendations in Table I of Dolan and Boomsma (1998) and were confirmed by us (see Comparison of Sampling Designs section) to be sensible design choice.

To study type I error, we used 10,000 datasets, and to study power we used 1,000 datasets. We simulated datasets of 100 families for the EDAC samples, and 250 families for the MDAC samples. The nominal type I error rate was set at .01. Marker data was simulated using eight equally-frequent alleles, with the marker at recombination fraction $\theta = 0$ for the power study and $\theta = 1/2$ for the type I error study.

As previously mentioned, some of the statistics (e.g., SCORE3) require that trait parameters be specified. When mapping is done with selected samples, population parameter values may not be available, and must be guessed or adopted from previous studies in other populations. We performed sensitivity studies to examine the robustness of the power of the statistics when there is misspecification of the trait parameters. We used models 1 and 1' only for this. We varied one parameter at a time while holding the other two parameters at the correct population values. Sibling correlation was set at 0.1 and 0.5, trait variance at values ranging from half the true value to twice the true value, and trait mean at the true mean plus and minus one to two standard deviations.

We implemented the following weights for our composite statistics, based on the strategy described in the previous section. For the RDP composite, we implemented (-0.726, 0.688), (-0.772, 0.635), (-0.815, 0.579), (-0.889, 0.458) as the weights (w_1, w_2) for the four ascertainment schemes, EDAC-3corner, EDAC-4corner, MDAC-3corner, MDAC-4corner, respectively. For the 3-part composite, we implemented (-0.239, -0.685, 0.688), (-0.254, -0.729, 0.635), (-0.436, -0.689, 0.579), (-0.462, -0.759, 0.458) as the weights (w_1, w_2, w_3) for the four ascertainment schemes.

7.5.2. Simulation results

7.5.2.1. Type I error Table 27 shows the type I error of each statistic considered in this study, based on 10,000 simulated datasets of EDAC-3corner samples for models 1-3, 1' and 2' only. Results for models 4-9 and for the other three types of samples were very similar to those shown here. Statistics that require trait parameters to be specified were computed with correct population values. Over the 10,000 simulated datasets, all statistics had a mean of zero; all except the Gu et al. (1996) IBD-sharing statistic had variance of one and approximately correct

type I error. (The 95% confidence interval for an estimated error rate of 1.00% is approximately 0.80% to 1.20%.) For the statistic HE-COM-correlation, type I error tended to be very slightly higher than 1%; whereas, for the IBD-sharing statistic Gu-empirical.variance, type I error tended to be slightly less than 1%. In general, the statistics SCORE3 and 3-part composite have the closest to correct type I error across all models.

7.5.2.2. Power Tables 28 through 31 give the power under all models for EDAC-3corner, EDAC-4corner, MDAC-3corner and MDAC-4corner samples respectively. Again, all the statistics were computed with their known population values. The 95% confidence interval for a power estimate of 50% is approximately 47% to 53%. For each model in each table, the power is shown in bold italics if it is within 5% of the highest power for that model.

The most important result is that the combination statistics have the highest power in general, although the correlation-based statistics do perform fairly well for some models, which is unlike what we observed with discordant sibling pair samples (chapter 5). Overall, using true population parameters SCORE3 is the best statistic for all of the four types of samples across all of the normal models (model 1 through model 9). This is consistent with theory and with previous studies of this statistic (chapter 5). In contrast, the two composite statistics are based on fixed external weights, and so they do not retain consistently high power across all models. It should be noted, however, that the 3-part composite, which does not depend at all on normality assumptions, has the highest power against both model 1' and model 2'. In fact, even the IBD-sharing statistic Gu-empirical.variance outperforms SCORE3 in many cases for the non-normal models. These results suggest that the most "nonparametric" statistics may be best for non-normal trait models.

7.5.2.3. Sensitivity To assess the robustness of the statistics to misspecification of the trait parameters, we carried out sensitivity analyses using parameter values that are guessed with error. We investigated the effect of misspecifying one parameter at a time. For each run, we set two of the parameters to the population values, and set the third to an arbitrary "wrong guess" (see methods). We performed these simulations on the same two sets of 1000 datasets, one from model 1 and one from model 1'. Tables 32 and 33 present these results using the EDAC sampling schemes. Results obtained with MDAC schemes had similar patterns and are not included here. The type I error was correct for all of these sensitivity studies (results not shown). Parameter misspecification does not affect Gu-empirical.variance, Haseman-Elston, RDP composite and 3-part composite, because they do not use the parameter estimates. On the other hand, for statistics HE-COM-correlation and SCORE3, misspecification of the population means has a large effect on their power. When correlation is misspecified, the impact on power is larger for SCORE3 than for HE-COM-correlation. Finally, wrong variances have little impact because this parameter has little contribution to non-centrality parameters of all the statistics examined. It is clear that the statistics RDP composite and 3-part composite have power that is most robust to parameter misspecification. However, this must be balanced against the lack of robustness of these statistics caused by the arbitrariness of the weights.

7.5.3. The bottom line

The IBD-sharing statistic proposed by Gu et al. (1996) is no longer the optimal choice for EDAC-type samples. Given a sensible EDAC design, SCORE3 should be the best statistic under most circumstances. For investigators who work with substantially non-normal models or who have poor estimates of population parameters, one of the composite statistics could serve as a good alternative.

7.6. COMPARISON OF SAMPLING DESIGNS

Earlier in this chapter, we reviewed EDAC design choices and outlined several issues that need to be reanalyzed in light of the availability of new test statistics. Having now established that SCORE3 is probably the best test statistic for almost any EDAC-type design, we can return to the question of design choice. We consider the following questions. 1) Is the recommended 2:1 ratio of discordants to concordants still the right choice if SCORE3 is used instead of an IBD-sharing statistic? (2) How do various EDAC and non-EDAC designs compare in terms of genotyping and phenotyping efficiency? (3) When is extreme sampling advantageous, and when is moderate sampling better?

7.6.1. Ratio of discordants to concordants

To look at this issue, we conducted simulation studies using three-corner sampling. We set thresholds for discordant pairs at 12% and 24%, and then varied the thresholds for concordant pairs. We fixed the number of families at 100 for samples with discordant threshold of 12% and at 250 for samples with discordant threshold of 24% (thus the genotyping sample sizes are 200 and 500, respectively.) We report power against model 1 for three statistics in table 34. These results show that as the discordant:concordant ratio varies, the power using the IBD-sharing statistic varies considerably. That is, for the IBD-sharing statistic it is important to get the ratio just right in order to maximize power. With SCORE3, it is less critical to get exactly the right ratio, as long as there is a preponderance of discordants. For example, when we select 250 pairs using 24% as the cut-off for the discordants, SCORE3 has consistently high power even as the ratio of discordants to concordants ranges from 10:1 to 2:1. The ratio issue interacts with the

issue of extreme or moderate sampling, in that it appears to be advantageous to collect more and more discordant pairs as the selection threshold for the *discordants* is relaxed.

If we only focus on the power results in table 34, we can almost conclude that any EDAC design that yields predominantly discordant pairs is good choice as along as SCORE3 is applied. However, if phenotyping numbers (listed in the last column of table 34) are also taken into account, it becomes apparent that that there are substantial savings associated with the "optimal" ratio. This question will be more extensively explored in the next sub-section.

7.6.2. Genotyping and phenotyping costs

Comparing different sampling designs is an important but complex task. For any real study, there will be a whole host of practical considerations that affect (and perhaps prevent) quantitative optimization. Genotyping and phenotyping costs will vary from study to study, as will absolute limits on the number of pairs available for screening. Some studies may start with no probands at all, while others may have a group of phenotyped probands or pairs already available. We approach the issue of design choice by considering a few relatively straightforward situations, and calculating screening, phenotyping and genotyping numbers for different sampling designs. Because of the computational requirements of this task, we report results only for model 1. While this does not answer every possible design question, it gives general guidance that can be adapted to many types of studies.

The first scenario we consider is the situation in which no individual has been recruited or phenotyped before the study begins. For this situation, table 35 and figure 6 (on page 158, 159) report screening, phenotyping, and genotyping numbers for various design options, each of which yields 80% power against model 1. To obtain the maximum accuracy (Yu et al. 2004), we calculated power by simulation as we did when comparing statistics. We used SCORE3 for all

calculations. Table 35 lists various design options, for example discordant pairs with a 10% cutoff, and reports the number of pairs screened, and the numbers of individuals phenotyped and genotyped for that option. Figure 6 plots smooth curves based on the genotyping and phenotyping numbers in table 35.

There are six different designs that we look at in table 35 and figure 6, each at varying extremity of selection. The six designs are as follows. 1) "EDAC-equal.threshold" is an EDAC design that uses the same percentile threshold to select both discordant and concordant pairs. This strategy should sample far more concordant than discordant pairs when the selection threshold is extreme, and close to equal numbers of the two types of sib pairs when the threshold is as moderate as 35%. 2) "EDAC-predominantly.discordant" uses a concordant threshold approximately 1/3 or more of the discordant threshold in most cases, in order to keep the ratio of discordant to concordant pairs close to 2:1. This line is meant to approximate the best one could do with EDAC sampling. 3) The third sampling scheme is discordant pairs only. 4) "Single.proband-both.sibs" phenotypes both siblings in each pair and genotypes any pair in which at least one sibling exceeds the threshold. 5) "Single.proband-one.sib" phenotypes only the first sibling in each pair, and then includes in the study only those pairs for which that first sibling exceeds the threshold. As compared to single.proband-both.sibs, this scheme should substantially reduce phenotyping, but at the cost of increasing the number of pairs screened. 6) The final design is population sampling, which is a single point on the plot. Bivariate scatter plots of each of these six types of samples from trait model 1 are shown in figure 5 on page 156.

If we ignore screening numbers (i.e. assume that we have as large a population as we want to choose pairs from), figure 6 contains all the information we need. Figure 6 can be interpreted as follows. The *x*-axis shows genotyping sample size, and the *y*-axis shows

phenotyping sample size. For a fixed phenotyping number, one could pick the design strategy that gives the lowest genotyping number. Conversely, one could fix the genotyping sample size at a desired level and choose the strategy with the lowest phenotyping requirement. If a line is below another at all values, we can say that it is a better sampling strategy. It is evident from figure 6 that discordant pairs, EDAC-equal.threshold, and single.proband-both.sibs are never the optimal choice. The discordant pair line does not even approach the population-sample point smoothly, because a discordant design excludes the very informative highly concordant pairs at every selection level. If phenotyping is cheap and we want to keep genotyping to a minimum, we should use an EDAC-predominantly.discordant strategy. If phenotyping is more expensive and genotyping cheaper, we should use single.proband-one.sib ascertainment. The crossover point *for this particular trait model* is at about 350 people (175 pairs) genotyped.

Screening comes into consideration generally in terms of absolute limits rather than costs, so the way to incorporate screening is to re-imagine figure 6 with any portions of lines that represent too-high screening numbers eliminated. For example, if we eliminate from figure 6 all points that require screening more than approximately 3000 pairs, we get figure 7 (on page 159). Under this tight screening limit, figure 7 clearly suggests different strategies. First, the minimum genotyping number is limited. For this particular trait model, with 3000 people screened, we can only get genotyping down to about 350 people while still retaining 80% power. Second, in order to reduce genotyping sample size to as low as possible, one should use the single.proband-both.sibs strategy – a design that is *never* recommended when screening is unlimited (see figure 6.) Third, the single.proband-one.sib strategy becomes the best choice for genotyping numbers above about 600, in contrast to the crossover point of about 350 seen in figure 6.

It was not possible to include optimal sampling in our simulation studies due to computation complexity, but we can comment on approximately where that line would fall in figure 6. Optimal sampling should be somewhat similar to EDAC-predominantly.discordant, but the genotyping sample size should be slightly less, and the phenotyping sample size equal to the total number of individuals (twice the number of pairs) screened. For more extreme sampling, there is probably not much difference in genotyping sample size between optimal sampling and EDAC-predominantly.discordant, but optimal sampling will require much more phenotyping. At the moderate sampling end of the curve, optimal sampling will probably have a stronger advantage over both EDAC-predominantly.discordant and single.proband-one.sib, and moreover, the difference in phenotyping will be less. Thus we expect optimal sampling to be less useful than other methods at the extreme sampling end of the spectrum, but potentially competitive at the moderate sampling end.

In a situation where some phenotyping has already been done before a study starts, the numbers in figure 6 and table 35 are not the right ones to use. If a population sample of sibling pairs has already been phenotyped, then optimal sampling is clearly the best choice - the only thing that must be decided is how deep to sample, which would be determined by the desired power. If a sample of pairs has already been phenotyped but it is not a population sample, the problem is somewhat more complicated, but optimal sampling should still be best if there is reasonable knowledge of the population trait parameters.

One of the most important situations to consider is that in which a sample of single probands has already been ascertained, as when an epidemiological or clinical study is expanded to become a family study. In this case, single-proband sampling is very efficient since it requires little new phenotyping, but EDAC designs still have the potential to be useful if it is desirable to

keep genotyping numbers down. For example, suppose we already have a population sample of 3000 unrelated probands phenotyped, and we want to decide the best strategy for sampling their siblings. The relevant numbers can be extracted from table 35 by considering the number of pairs screened to be fixed at 3000, and subtracting the already-phenotyped probands from the phenotyping numbers. If we use the single.proband-one.sib approach (10% threshold), table 35 tells us we need to phenotype an additional 300 individuals and genotype 600. If we use an EDAC-predominantly.discordant approach (24%, 8%), we need to phenotype an additional 1419, but only genotype 480. The worst case is discordant sibling pairs (50% threshold), which require phenotyping an additional 2964 and genotyping 2400. If we assume we start with approximately 6000 phenotyped individuals instead of 3000, the single.proband-one.sib approach (4% threshold) requires phenotyping an additional 138 and genotyping 276, whereas the EDAC-predominantly.discordant approach (12%, 4%) requires phenotyping 1395 and genotyping 194. Thus for a larger initial population and extreme selection single.proband-one.sib could be the best choice, but for a smaller population and if genotyping cost must be minimized, an EDAC strategy might be useful.

7.6.3. Extreme or moderate sampling

Figure 6 can also be used to think about extreme vs. moderate sampling. In that plot each line was generated by varying extremeness of thresholds; as the selection thresholds are relaxed, the lines are extended to the right until reaching the point of population sampling. As the genotyping number increases to 500, there is a sharp downward slope for each line, which indicates that moderate sampling substantially reduces phenotyping sample size for fixed statistical power. However, when genotyping sample size is larger than about 600 all lines plateau, indicating little benefit to very moderate selected sampling of any type, except that screening numbers continue

to go down throughout this region (table 35). The precise forms of these lines will be highly dependent on the trait model, but we suggest that one should not use a threshold higher than 35% for discordant pairs (with or without concordants), or a threshold higher than 15% with single-proband sampling unless it is necessary to reduce screening numbers.

7.7. DISCUSSION

We reviewed available design and statistic choices for EDAC sibling pair studies, and investigated what statistics are most powerful for each design. We concluded that the IBD-sharing statistic (Gu et al. 1996) is no longer the best choice, and that new combination statistics are most powerful for EDAC-type samples. When trait parameters are known, the best statistic is SCORE3; when the estimates of trait parameters are doubtful or when trait models are substantially non-normal, our 3-part composite statistic can be a good alternative.

We then used SCORE3 to compare the power of various EDAC and non-EDAC design choices. We showed that a "sensible" EDAC design means one in which we sample significantly more discordant pairs than concordant pairs. We concluded that in the situation where no phenotyping/enrollment has been done ahead of time, a sensible EDAC design is most powerful as well as most cost effective when genotyping numbers must be low. For higher genotyping and lower phenotyping requirements, single-proband ascertainment is probably preferable. We also noted that the use of combination statistics gives the traditional EDAC design more flexibility by allowing moderate sampling and by lessening the need use exactly the right ratio of discordant to concordant pairs. When single probands have already been phenotyped, it is probably most efficient to use a single-proband design, unless it is desirable to get genotyping numbers as low

as possible. If a sample of pairs has already been phenotyped, clearly optimal sampling is the best choice.

Our SCORE3 statistic is not unique. There are several other statistics in the literature that are very similar and should perform equivalently for sibling pair data. One is the HE-COM-combination statistic (Sham and Purcell 2001), and another is the ascertainment-corrected variance components statistic proposed by Sham et al. (2000), which conditions on trait values. Yet another is the robust regression-based statistic proposed by Sham et al. (2002), which regresses IBD on trait values. Finally, Chen et al. (2004) discuss a generalized estimating equation (GEE) framework that unifies a number of the methods we have discussed.

Our results are basically consistent with those in previous studies. We did not consider the IBD-sharing statistic variants proposed by Li and Zhang (2000) and Li and Gastwirth (2001). Given that these statistics use IBD-sharing information only, we do not expect them to perform better than the combination statistics for EDAC sibling pairs. Our finding that combination statistics are powerful for multiple-proband sampling (samples collected on the basis of two or more people in each pedigree having particular phenotypes) was first noted by Forrest and Feingold (2000), and has also been discussed Szatkiewicz et al. (2003) and Szatkiewicz and Feingold (submitted) (e.g., chapters 5 and 6). As for EDAC design strategies, our results agree with most previous conclusions (e.g., Gu et al. 1996, Gu and Rao 1997, Dolan and Boomsma 1998). Among the literature comparing different sibling pair strategies, Gu et al. (1996) and Gu and Rao (1997) showed that an EDAC design is more cost effective than using discordant pairs only, and many studies (e.g. Risch and Zhang 1995, Purcell et al. 2001) showed that multiple-proband sampling is more powerful than single-proband sampling per genotyping. Using the more powerful statistic SCORE3, we found that single proband sampling and moderate sampling

can be useful as well when phenotyping cost and screening limitations are also taken into account.

Our comparison of different sampling designs has some obvious limitations. Most notably, we used samples simulated under trait model 1 only. While exploring the ratio issue for EDAC designs, we found that the optimal ratio of discordants to concordants is fairly dependent on the trait model. To search for the optimal values, one could in principle derive an optimization algorithm for fixed analytical power on the basis of trait values and model parameters; however, that "optimal" ratio will only be specific for one trait model and will likely be sensitive to parameter misspecification. We believe that, given the recommended combination statistics applied, under just about any reasonable trait model, an EDAC sample with predominantly discordant pairs should perform fairly well.

There are also some limitations of the models we studied. All of our models used an environmental/polygenic sibling correlation of .25. The size of this covariance will mostly affect the informativeness of the concordant pairs in an EDAC sample. This should not affect the relative power of all the statistics we considered. It does slightly affect the weights to be implemented in composite statistics. In general, the smaller the correlation, the larger the weights associated with concordant pairs. However, we found that even under a model with correlation of 0, there is no significant information obtained in the actual trait values among the concordants.

An additional limitation in the trait models we studied is that our non-normal models 1' and 2' are generated by ad-hoc transformation. Perhaps the most important open questions about trait model are what types of non-Gaussian trait models are most realistic and important, and how various features of the models and statistics interact to determine power. The fact that our results for models 1' and 2' were significantly different from our results for mixture-of-normals

models indicates the need for further exploration. Although all of our discussion has focused on sibling pairs, our results have a number of implications for studies using larger pedigrees. Several recent papers showed that it is more useful to collect whole sibships than extreme pairs only (e.g. Alcais and Abel 2000,Tang and Siegmund 2001, Purcell et al. 2001). With larger sibships, the design comparison issues will be very different and will require an entirely new study. For example, one of our results was that discordant-pair only designs are not a good idea. But with larger sibships it might be sufficient to ascertain sibships containing a discordant pair, since some concordant siblings would end up being included as well. On the statistical methods issues, our results should be very relevant to larger sibships. For multiple-ascertainment of larger sibships, combination statistics will be essential as they are for sibling pairs. Of the statistics we considered, the score statistic extends naturally to larger sibships and it is likely to be the best choice. The methods proposed by Sham et al. (2002) and Chen et al. (2004) were developed particularly for extended pedigrees, and they are also good candidates.

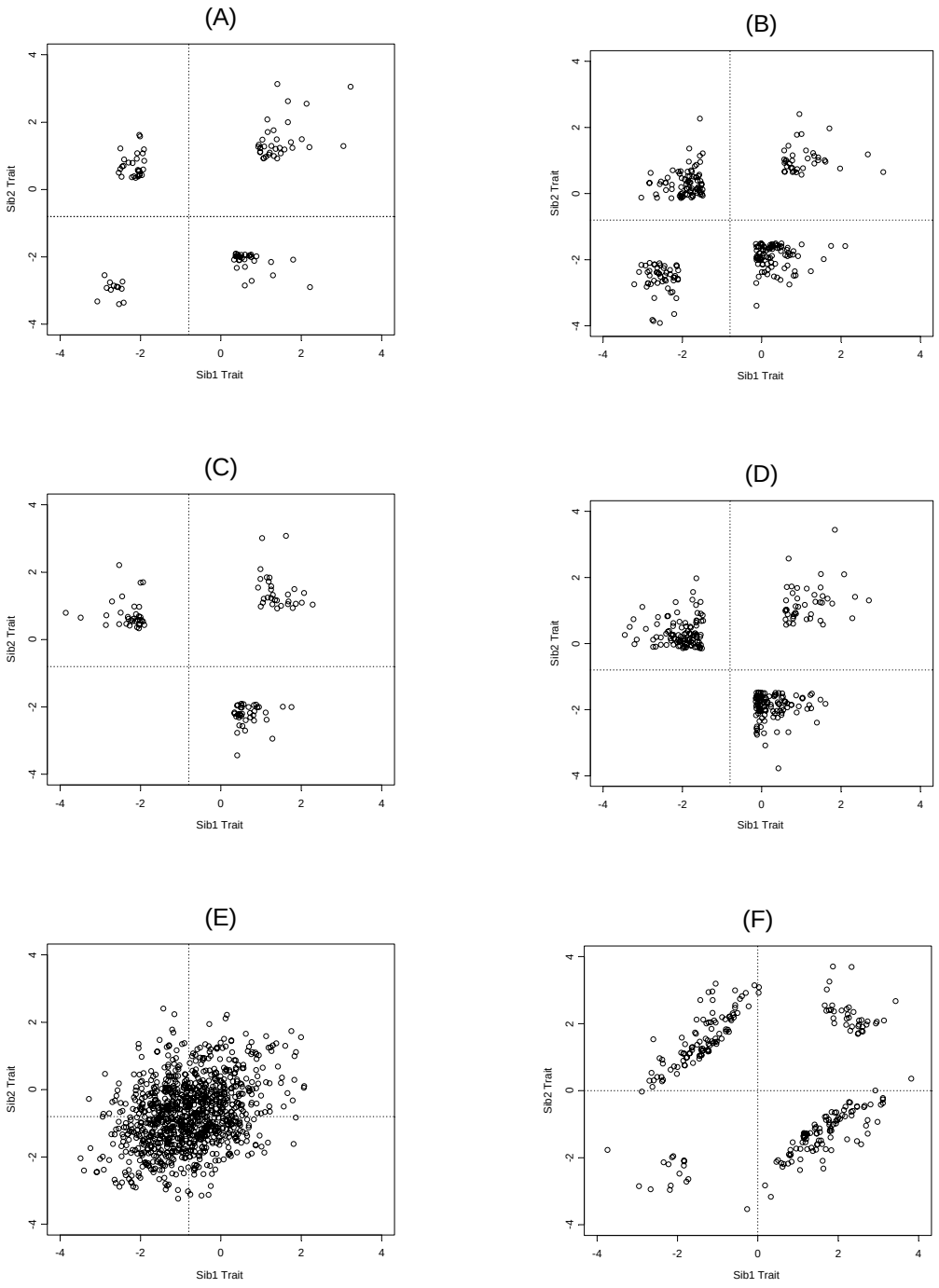

Figure 4 Scatter plots of simulated EDAC samples from trait model 1

A) EDAC-4corner, discordant at 12%, concordant high and low at 4%, n=100
B) MDAC-4corner, discordant at 24%, concordant high and low at 8%, n=250
C) EDAC-3corner, discordant at 12%, concordant high at 4%, n=100
D) MDAC-3corner, discordant at 24%, concordant high at 8%, n=250
E) Population sample, n=1500;
F) 5% most informative sib pairs, n=250

Table 26 Genetic models – chapter 7

	Value for model										
PARAMETER	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1'**	**2'**
Model-defining											
Type of inheritance[a]	Add	Dom	Rec	Add	Dom	Rec	Add	Dom	Rec	Add	Dom
Locus heritability	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	0.2	N.A.[b]	N.A.
Allele frequency	0.1	0.1	0.1	0.5	0.5	0.5	0.9	0.9	0.9	0.1	0.1
Trait means	-1, 0, 1	0, 1, 1	0, 0, 1	-1, 0, 1	0, 1, 1	0, 0, 1	-1, 0, 1	0, 1, 1	0, 0, 1	-1.6, 0, 1.6	0, 1.6, 1.6
Environmental SD	0.849	0.785	0.199	1.414	0.866	0.866	0.849	0.199	0.785	N.A.	N.A.
Environmental correlation	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	N.A.	N.A.
Calculated											
Overall mean	-0.8	0.19	0.01	0.0	0.75	0.25	0.8	0.99	0.81	-1.32	0.295
Overall SD	0.949	0.877	0.222	1.581	0.968	0.968	0.949	0.222	0.877	2.047	1.393
Skewness	0.168	0.140	0.880	0.0971	-0.0991	0.102	-0.168	-0.880	-0.140	-1.587	1.504
Kurtosis	0.101	0.0240	3.802	0.0556	-0.0714	-0.031	0.101	3.802	0.0240	5.268	9.406
Overall correlation	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.25	0.26

a. Add = additive; Dom = dominant; Rec = recessive.
b. N.A. = not applicable.

Table 27 Type I error for EDAC-3corner samples

	Error (%) under model				
STATISTIC	1	2	3	1’	2’
Gu	0.63	0.54	0.55	0.56	0.56
Gu-empirical.variance	0.94	0.93	0.91	0.90	0.90
Haseman-Elston	1.10	1.07	1.01	1.06	1.09
HE-COM-correlation	1.15	1.24	1.14	1.12	1.02
RDP	1.00	1.06	0.89	0.90	0.90
SCORE3	1.09	1.05	0.94	0.94	0.95
RDP composite	0.99	1.03	0.95	0.95	0.84
3-part composite	1.00	1.04	0.95	0.98	0.98

Note- the nominal type I error rate was set at 1%

Table 28 Power for EDAC-3corner samples at α=1% level

	Power under model										
STATISTIC	1	2	3	4	5	6	7	8	9	1’	2’
Gu-empirical.variance	.74	.68	.58	.47	.21	.61	.23	.02	.22	***.75***	.69
Haseman-Elston	.77	.69	.53	.52	.27	.61	.33	.10	.28	.49	.25
HE-COM-correlation	***.82***	***.74***	***.94***	.55	.29	.66	.33	.08	.29	.74	.67
RDP	.49	.50	.05	.59	***.59***	.50	***.60***	.07	***.61***	.33	.27
SCORE3	***.79***	***.75***	.87	***.67***	***.58***	***.71***	***.59***	.09	***.58***	.71	.67
RDP composite	***.81***	***.76***	.59	.60	.42	***.69***	.42	.05	.39	***.76***	.71
3-part composite	***.81***	***.76***	.50	.59	.41	***.69***	.42	.04	.39	***.78***	***.77***

Note: Under each model, the highest power estimates and those retaining 95% of the highest power are labeled in bold italics.

Table 29 Power for EDAC-4corner samples at α=1% level

	Power under model										
STATISTIC	1	2	3	4	5	6	7	8	9	1’	2’
Gu-empirical.variance	.58	.53	.22	.55	.47	.49	.61	.27	.52	***.61***	.53
Haseman-Elston	.65	.56	.28	.59	.51	.52	.64	.32	.56	.36	.20
HE-COM-correlation	***.70***	***.59***	***.81***	***.62***	***.56***	***.55***	***.68***	***.84***	***.62***	.38	.54
RDP	.38	.42	.03	.47	.42	.40	.40	.04	.38	.20	.17
SCORE3	***.69***	***.61***	***.77***	***.63***	***.59***	***.56***	***.69***	***.80***	***.62***	.36	.52
RDP composite	***.66***	***.58***	.30	***.61***	**.56**	***.56***	***.65***	.35	***.59***	.60	.53
3-part composite	***.66***	***.58***	.24	***.61***	.55	***.56***	***.65***	.28	***.59***	***.64***	***.58***

Note: Under each model, the highest power estimates and those retaining 95% of the highest power are labeled in bold italics.

Table 30 Power for MDAC-3corner samples at α=1% level

	Power under model										
STATISTIC	1	2	3	4	5	6	7	8	9	1’	2’
Gu-empirical.variance	.65	.62	.15	.38	.19	.56	.13	.02	.18	.66	.60
Haseman-Elston	.71	.64	.22	.48	.32	.56	.32	.08	.34	.24	.14
HE-COM-correlation	***.79***	.76	***.70***	.55	.34	.66	.32	.08	.33	.54	.60
RDP	.58	.62	.07	.69	***.64***	.63	***.62***	.08	***.66***	.32	.27
SCORE3	***.81***	***.81***	.57	***.76***	***.66***	***.78***	***.61***	.08	***.65***	.61	.65
RDP composite	***.81***	***.81***	.20	***.72***	.57	***.77***	.52	.06	.55	.68	.72
3-part composite	***.81***	***.81***	.17	.71	.56	***.77***	.52	.05	.54	***.74***	***.80***

Note: Under each model, the highest power estimates and those retaining 95% of the highest power are labeled in bold italics.

Table 31 Power for MDAC-4corner samples at α=1% level

	Power under model										
STATISTIC	1	2	3	4	5	6	7	8	9	1’	2’
Gu-empirical.variance	.52	.53	.08	.54	.50	.50	.49	.07	.53	.54	.55
Haseman-Elston	.60	.59	.14	.59	.53	.50	.61	.17	.61	.23	.13
HE-COM-correlation	***.69***	.66	***.54***	.65	.60	.60	***.69***	***.54***	.67	.38	.59
RDP	.48	.54	.05	.59	.53	.49	.50	.07	.53	.27	.20
SCORE3	***.72***	***.71***	.49	***.71***	***.69***	***.67***	***.71***	.48	***.73***	.40	.59
RDP composite	***.69***	***.69***	.12	***.72***	***.69***	***.65***	***.68***	.12	***.72***	.55	.61
3-part composite	***.69***	***.69***	.10	***.72***	***.68***	***.65***	.67	.10	***.71***	***.60***	***.69***

Note: Under each model, the highest power estimates and those retaining about 95% of the highest power are labeled in bold italics.

Table 32 Sensitivity analysis varying means

EDAC-3corner samples

	Model 1					Model 1'				
STATISTIC	μ-2σ	μ-σ	μ	μ+σ	μ+2σ	μ-2σ	μ-σ	μ	μ+σ	μ+2σ
HE-COM-correlation	.80	.81	.82	.81	.66	.77	.77	.74	.54	.03
SCORE3	.66	.81	.79	.71	.56	.63	.75	.71	.49	.02
RDP composite[a]			.81					.76		
3-part composite[a]			.81					.78		
Gu-empirical.variance[a]			.74					.75		

EDAC-4corner samples

	Model 1					Model 1'				
STATISTIC	μ-2σ	μ-σ	μ	μ+σ	μ+2σ	μ-2σ	μ-σ	μ	μ+σ	μ+2σ
HE-COM-correlation	.68	.74	.70	.43	.15	.63	.64	.38	.11	.03
SCORE3	.56	.72	.69	.41	.13	.56	.61	.36	.09	.02
RDP composite[a]			.66					.60		
3-part composite[a]			.66					.64		
Gu-empirical.variance[a]			.58					.61		

a. Power of these statistics does not change as parameters vary.

Table 33 Sensitivity analysis varying correlation

EDAC-3corner samples

	Model 1			Model 1'		
STATISTIC	$r = 0.1$	$r = 0.3$	$r = 0.5$	$r = 0.1$	$r = 0.3$	$r = 0.5$
HE-COM-correlation	.82	.82	.81	.78	.74	.59
SCORE3	.81	.79	.68	.78	.71	.48
RDP-COMPOSITE[a]		.81			.76	
3part-COMPOSITE[a]		.81			.78	
Gu-robust[a]		.74			.75	

EDAC-4corner samples

	Model 1			Model 1'		
STATISTIC	$r = 0.1$	$r = 0.3$	$r = 0.5$	$r = 0.1$	$r = 0.3$	$r = 0.5$
HE-COM-correlation	.71	.70	.69	.28	.38	.42
SCORE3	.65	.69	.60	.24	.36	.35
RDP-COMPOSITE[a]		.66			.60	
3part-COMPOSITE[a]		.66			.64	
Gu-robust[a]		.58			.61	

a. Power of these statistics do not change as parameters vary.

Table 34 Comparing EDAC designs under trait model 1

Cutoff Discordant	Cutoff Concordant[a]	Ratio[b] DS:CS	Power of statistic			Sample size	
			SCORE3	Gu-empirical[c]	HE-COM-correlation	genotyping	phenotyping
12%	2%	4.6 : 1	.85	.66	.77	200	9835
12%	3%	2.5 : 1	.82	.74	.81	200	8540
12%	4%	**1.7 : 1**[d]	.81	.76	.83	200	7313
12%	6%	1:1	.74	.69	.78	200	5635
12%	9%	1:2	.63	.55	.64	200	4045
12%	12%	1:3	.54	.43	.53	200	3005
24%	4%	10 : 1	.81	.45	.67	500	5291
24%	8%	3.6 : 1	.82	.64	.79	500	4616
24%	10%	**2.6 : 1**[d]	.80	.66	.80	500	4228
24%	12%	2:1	.78	.62	.78	500	3856
24%	18%	1:1	.69	.51	.66	500	3050
24%	24%	1:1.5	.56	.36	.56	500	2380

Note – ratio, power of statistics are given over 1000 replicates.
a. Concordant pairs are sampled from the top tail only.
b. Ratio of discordant sibs to concordant sibs
c. It is the Gu-empirical.variance.
d. Numbers in bold indicate optimal ratio.

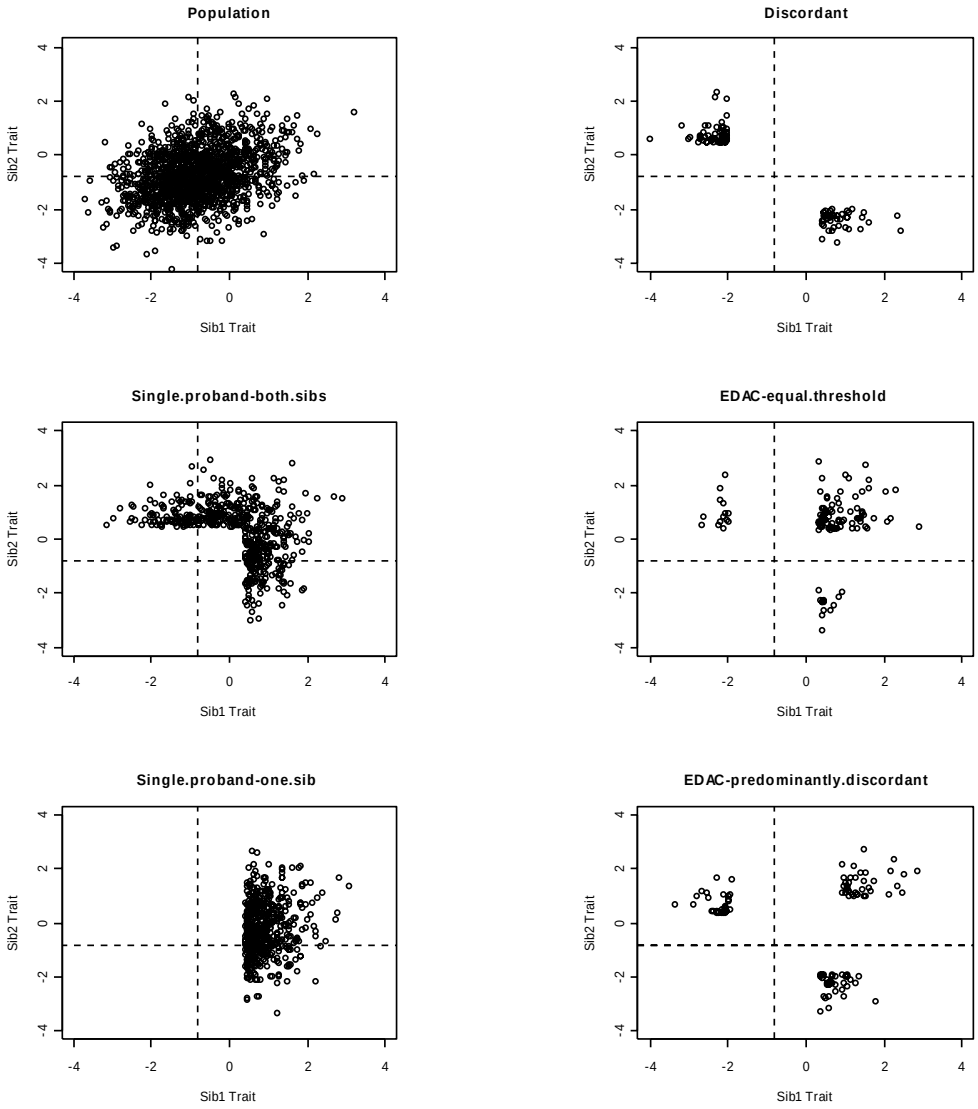

Figure 5 Bivariate scatter plots of six different sampling designs from trait model 1

Table 35 Average sample sizes required for 80% power at α=1% level under trait model 1

Type of sample	Thresholds[a]	Ave. Pairs screened	Ave. People phenotyped	Ave. People genotyped
EDAC-equal.threshold				
	6%,6%	7,528	8,842	200
	8%,8%	5,875	6,820	250
	12%,12%	4,164	5,170	360
	20%,20%	2,865	4,007	620
	30%,30%	2,264	3,623	1,100
EDAC-predominantly.discordant				
	6%,1%	17,337	19,417	88
	8%,2%	10,355	12,017	124
	12%,4%	5,831	7,226	194
	24%,8%	2,958	4,377	480
	30%,12%	2,423	3,881	700
	35%,16%	2,192	3,721	940
Discordant				
	6%	28,827	32,326	104
	8%	18,532	21,483	134
	12%	9,960	12,350	210
	18%	6,006	8,169	360
	35%	3,449	5,862	1140
	50%	2,964	5,928	2400
Single.proband-both.sibs				
	1%	5,862	11,723	230
	3%	3,122	6,243	354
	5%	2,453	4,905	450
	10%	1,973	3,945	700
	15%	1,869	3,737	966
	20%	1,801	3,601	1,200
Single.proband-one.sib				
	1%	9,801	9,904	206
	3%	6,100	6,238	276
	5%	4,010	4,212	404
	10%	3,040	3,340	600
	15%	2,934	3,374	880
	20%	2,759	3,309	1,100
Population sample				
	-	1,850	3,700	3,700

Note – all of the required sample sizes and the 80% power were obtained over 1000 replicates
a. For the thresholds listed for EDAC type samples, the first number is for selecting discordant pairs and the second number is for selecting pairs concordant for high values.
b. All the calculation used SCORE3.

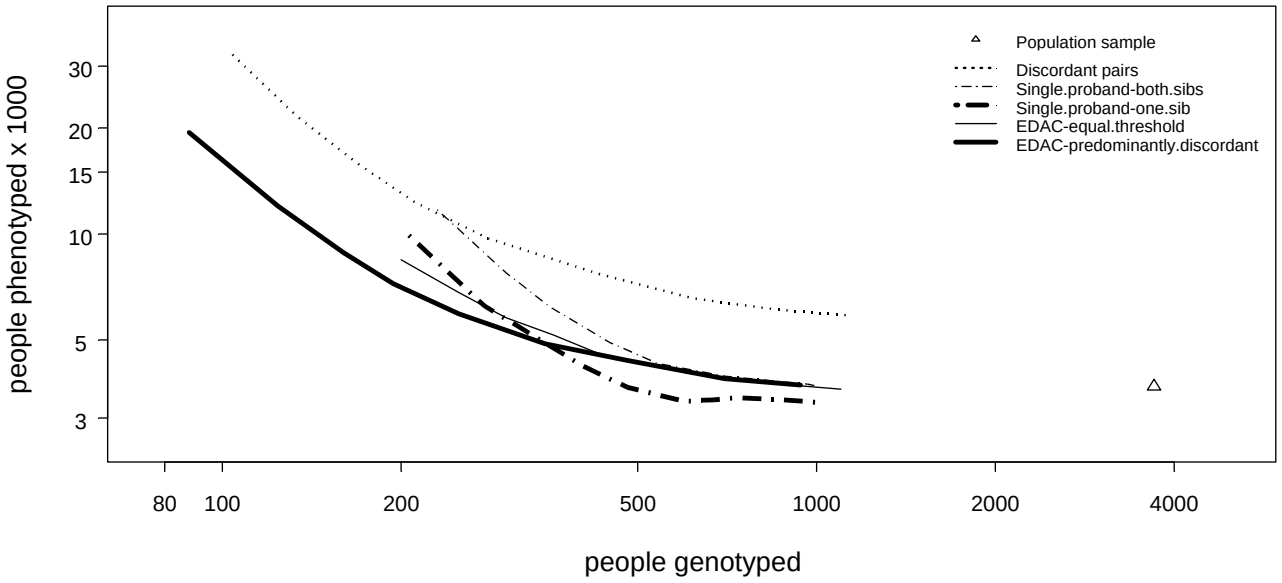

Figure 6 Plot of genotyping vs. phenotyping sample size for 80% power

Note – Results were obtained under trait model 1, when screening is unlimited.

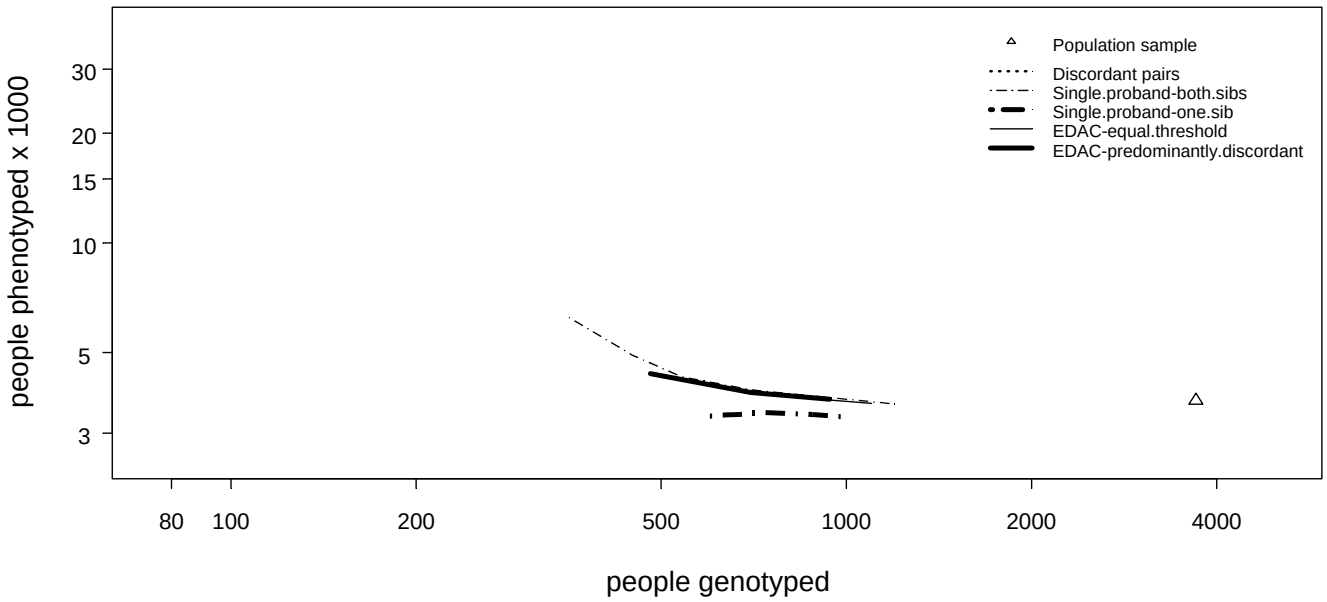

Figure 7 Plot of genotyping vs. phenotyping sample size for 80% power and limited screening

Note – Results were obtained under trait model 1, when screening is limited to <=3,000

8. QTL MAPPING IN CONCORDANT/AFFECTED SIBLING PAIRS

This chapter discusses our study of statistics for QTL mapping with concordant/affected sibling pairs ascertained on the basis of a binary disease trait. In chapter 7 we talked about concordant pairs in the content of EDAC (extreme discordant and concordant) design; here we consider concordant pairs alone. Concordant pairs are generally not considered to be very powerful but sometimes this is what one has, e.g. in an affected sibling pair study.

8.1. INTRODUCTION

8.1.1. Concordant sibling pairs

Concordant pairs were first discussed by Risch and Zhang (1995); they considered sibling pairs concordant for extreme values, e.g. pairs where both siblings are in the top 10% or both are in the bottom 10%. However this type of design was generally viewed to be not very useful (e.g. Risch and Zhang 1995, Gu et al. 1996, Allison et al. 1998). The reason is that the power of concordant pairs is heavily dependent on the trait model. In addition, when there is a positive residual correlation, the power of concordant pairs to detect the genes decreases. As it is usually not possible to know beforehand the underlying genetic models, Zhang and Risch (1996) examined further the use of extreme concordant pairs but considered the effect of parents' phenotypes. They recommended excluding from linkage studies the sib pairs whose parents also have similar extreme values.

We know if sibling pairs are concordant for extreme values, the IBD-sharing near the trait is substantially lower than the expected value under null hypothesis. Thus the mean IBD-sharing statistic was used in the studies of concordant pairs (Risch and Zhang 1995, Zhang and Risch 1996, Gu et al. 1996, Allison et al. 1998). This is also the same test statistic that is used for affected sibling pair mapping of binary disease traits (e.g., Blackwelder and Elston 1985).

8.1.2. Affected sibling pairs

Affected sibling pairs can be considered concordant pairs for any quantitative traits associated with the disease they were originally collected to study (e.g. glucose and insulin levels for diabetes). Recently, there has been substantial interest in using affected sibling pairs from binary-trait linkage studies to map QTLs as a tool for finding the disease associated genes. For example, Fisher et al. (1999) mapped a possible dyslexia locus by focusing on several different quantitative traits that are correlated with different components of the dyslexia phenotype. Similar approaches have been applied to type II diabetes (e.g. Watanabe et al. 2000), schizophrenia (e.g. Cai et al. 2001), extreme obesity (Dong et al. 2003), etc.

For most of those studies using affected sibling pairs to map QTLs, linkage analysis was done with correlation-based statistics (e.g. Watanabe et al. 2000, Cai at al. 2001, Dong et al. 2003). Huang and Jiang (2003) recently proposed a likelihood-based statistic for incorporating quantitative trait information into an affected sibling pair linkage analysis. But it is not clear how this statistic compares to QTL mapping combination statistics.

Using similar methods to our discordant pair studies (chapter 5), here we seek to understand what are the best statistical methods for concordant/affected sibling pairs, and what are the appropriate design strategies.

8.2. METHODS

8.2.1. Simulation for affected sib pairs

In addition to strictly concordant sibling pairs, we added what we call "fuzzy ascertainment" to our simulation program. The idea is that when affected sibling pairs are collected for a binary disease, they can be thought of as concordant pairs for any quantitative trait associated with the disease. Under a simple model where the quantitative trait *is* the liability for the disease, the ascertainment scheme with respect to the quantitative trait is then quite precise. But under a more realistic complex trait model, we have ascertained only approximately concordant pairs for the quantitative trait. We implemented this in our simulation software by generating two different quantitative traits for each person, with a user-specified level of correlation between them. This allows us to ascertain on one of the traits (think of it as the liability), but then perform the QTL mapping with the other.

8.2.2. Composite statistic for concordant pairs

We have developed a composite statistic for concordant/affected sibling pairs. We found that the most powerful composite statistic was a weighted sum of the IBD-sharing statistic and HE-COM-correlation of Sham and Purcell (2001), i.e.:

$$w_1(HE.COM.correaltion) + w_2 IBD2 .$$

HE-COM-correlation is used because it is the most powerful among all correlation-based statistics for concordant/affected sib pairs. This also means that this composite statistic is not parameter free. As with our previous composite statistics, we performed simulation to explore sensible weights (w_1, w_2). We found that the weights depend primarily on the ascertainment scheme, not on the trait model. Thus the weights were implemented for each ascertainment scheme we considered (see next section.)

8.2.3. Simulation studies

We simulated strictly concordant sibling pairs using both extreme and moderate selection thresholds. We performed a simulation study similar to those in our discordant pairs work to test the various statistics for this type of sample. The type I error results are consistent with those in previous studies (chapters 4 and 5) and thus are omitted here. Power results are shown in table 36 and 37. As would be expected from our previous work, the score statistic, SCORE3, was most powerful if parameter values were known. The composite statistic was not completely parameter free, but it was more robust to parameter misspecification than the score statistic. However, under many models, even the best statistics had very little power. This is certainly consistent with popular wisdom, and it suggests that the power of strictly concordant sib pair samples is limited.

Although we did not conduct simulation studies particularly with affected sibling pairs, we expect too see that the power of affected sibling pairs also heavily depends on the relationship between the binary and quantitative traits. However, the fuzzy ascertainment spreads out the sample of pairs (i.e., draws some pairs from places other than the "highly concordant" region of the joint distribution), so this type of sample is expected to be more powerful.

8.3. CONCLUSION AND DISCUSSION

For concordant/affected sibling pairs, we concluded that the best statistic is combination statistic SCORE3 when correct population trait parameters are available. But the power of both concordant and affected sibling pairs is heavily depends on trait models. We also expect that the affected sib pairs can be more useful than strictly concordants.

We recommend further study of designs that use affected sibships that are already collected for linkage studies of binary traits. There is plenty of evidence that larger sibships are more powerful than extreme sibling pairs, even in the context of discordant sib pair design (e.g. Alcais and Abel 2000, Tang and Siegmund 2001). Recently, some real studies used affected sibships to map QTLs (e.g. Alarcon et al. 2002, Wiltshire et al. 2002). We expect affected sibships to give more information than affected sibling pairs. When additional siblings are available besides the affected pairs in the family, one can either include the whole sibships (recruit both affected and unaffected siblings) or include only the affected siblings in the QTL study. It is not entirely clear when is advantageous to use which strategy. Of all the statistics we considered, the score statistics extend in a natural way to larger sibships and are probably the logical choice. One problem that arises when a mix of pedigree types is used is that there may no longer be a good way to calculate an empirical variance of the IBD sharing in order to form a statistic that has the correct type I error. Further investigation is warranted.

Table 36 Power for extremely concordant sibpair samples at α=1% level

	Models										
	1	2	3	4	5	6	7	8	9	1’	2’
Group A											
IBD2	0.72	0.69	0.38	0.22	0.07	0.52	0.04	0.01	0.05	0.57	0.50
IBD3	0.72	0.69	0.38	0.22	0.07	0.52	0.04	0.01	0.05	0.57	0.50
Group B											
ORIGINAL.HE	0.01	0.01	0.08	0.01	0.01	0.01	0.02	0.01	0.01	0.01	0.01
TRAIT.SUM	0.09	0.03	0.985	0.02	0.01	0.03	0.01	0.01	0.01	0.05	0.05
TRAIT.PRODUCT	0.11	0.04	0.996	0.02	0.01	0.03	0.01	0.01	0.01	0.07	0.05
XU	0.07	0.02	0.779	0.02	0.01	0.02	0.01	0.01	0.01	0.06	0.03
HE-COM-correlation	0.12	0.05	0.998	0.03	0.01	0.04	0.01	0.01	0.01	0.09	0.04
Group C											
HE-COM-combination	0.81	0.73	0.998	0.23	0.07	0.56	0.04	0.01	0.05	0.65	0.35
SCORE3	0.79	0.72	0.998	0.21	0.06	0.54	0.04	0.01	0.05	0.63	0.33
COMPOSITE2*	0.71	0.56	0.996	0.15	0.05	0.42	0.03	0.01	0.04	0.58	0.39

Note: * Equal weights are implemented for COMPOSITE2 for both extremely and moderately concordant sibling pairs.

Table 37 Power for moderately concordant sibpair samples at α=1% level

	Models										
	1	2	3	4	5	6	7	8	9	1’	2’
Group A											
IBD2	0.26	0.33	0.02	0.28	0.20	0.40	0.12	0.13	0.13	0.35	0.29
IBD3	0.26	0.33	0.02	0.28	0.20	0.40	0.12	0.13	0.13	0.35	0.29
Group B											
ORIGINAL.HE	0.08	0.06	0.30	0.02	0.02	0.04	0.01	0.01	0.02	0.01	0.02
TRAIT.SUM	0.27	0.26	0.53	0.06	0.01	0.17	0.01	0.01	0.01	0.11	0.14
TRAIT.PRODUCT	0.38	0.34	0.76	0.08	0.02	0.23	0.01	0.01	0.01	0.21	0.25
XU	0.37	0.27	0.93	0.07	0.02	0.17	0.02	0.01	0.01	0.25	0.22
HE-COM-correlation	0.53	0.45	0.96	0.11	0.03	0.31	0.01	0.01	0.01	0.33	0.23
Group C											
HE-COM-combination	0.71	0.70	0.93	0.38	0.18	0.66	0.10	0.01	0.11	0.63	0.31
SCORE3	0.71	0.70	0.93	0.38	0.18	0.65	0.10	0.01	0.11	0.63	0.30
COMPOSITE2*	0.72	0.71	0.76	0.36	0.15	0.65	0.08	0.01	0.08	0.65	0.51

9. CONCLUSION AND DISCUSSION

9.1. SUMMARY

Selected samples are far more common than population samples in human QTL mapping studies, however little was known between the interplay of statistical methods and selected sampling before this work. This dissertation focuses on sibling pairs and seeks to fill this gap in knowledge by two constituent steps. The first is to know what are the best statistics for each type of selected sib pair sample. The second is to understand what kinds of sampling are advantageous in what situation given the best statistics.

To answer the first question, we considered various sampling schemes that use sibling pairs. Chapter 4 looks at both population sampling and single-proband sampling; chapters 5 and 6 focuses on extreme discordant (EDSP) and moderately discordant sibling pairs (MDSP); chapter 7 considers combined extreme discordant and concordant (EDAC) sib pairs of various types; and chapter 8 explores concordant/affected sibling pairs. With each type of sampling scheme considered, we performed the most comprehensive comparison to date of statistics for QTL mapping using analytical and simulation approaches. We found that only some statistics have consistently correct type I error over all the models and selected sampling schemes we considered, and that the statistics that give incorrect type I error should not be recommended for selected samples. We identified the most powerful statistics for each sampling scheme, considering the situations of perfectly known and of misspecified trait parameters. We noted that

parameter misspecification is a much more important issue for selected samples; and we found that even among the methods that are most powerful for "nice" data, some are more robust than others to non-Gaussian trait models and/or misspecified trait parameters.

The specific conclusions about the best statistics for each sampling scheme considered are summarized in tables 38 and 39 and are interpreted below. (1) For both population and single-proband samples, the statistics having highest power are XU (Xu et al. 2000), HE-COM-correlation, HE-COM-combination (Sham and Purcell 2001) and SCORE3; with XU having robust power against the non-normal trait models we studied. When the effect of misspecification of parameters is taken into account, XU is the most robust for population sample, whereas HE-COM-correlation becomes the most robust for single-proband sample. (2) For any of the multiple-proband sampling schemes considered (EDSP, MDSP, various EDAC, concordant/affected), it is very important to use a combination statistic to draw maximum power. In general, statistics SCORE3 and HE-COM-combination are most powerful if parameter values are known; and the composite statistics are more robust to parameter misspecification and to non-normal trait models. The robust discordant pair (RDP) statistic is the best choice for both EDSP and MDSP, but it is only appropriate for discordant pairs. The traditional IBD-sharing statistics are limited and are powerful only with very extreme selection criteria. (3) Overall, SCORE3 and HE-COM-combination are the universal most powerful statistics for any type of sibling pair sample given well-known trait parameters, and they do not require the knowledge of the exact ascertainment scheme.

Table 38 The most powerful statistics for various sibling pair samples when model parameters are well known

	Type of samples from mixture-of-normals model						
BEST STATISTICS	Population sampling	Single-proband sampling	Multiple-proband sampling				
			EDSP	MDSP	Concordant[3]	EDAC[3]	Non-standard designs[4]
Correlation-based							
XU	+	+					
HE-COM-correlation	+	+					
IBD-sharing							
IBD-empirical.variance[1]			+				
Combination							
HE-COM-combination	+	+	+	+	+	+	+
SCORE3	+	+	+	+	+	+	+
RDP			+	+			
Composite[2]			+	+	+	+	

Note – statistics in *Italics* are more robust against non-Gaussian trait models

1. IBD-sharing statistics with empirical variance – It refers to IBD2 for discordant/concordant pairs; and it referes to Gu-empirical.variance for EDAC pairs.
2. Composite statistics should be appropriately weighted according to type of samples.
3. Including both extreme sampling and moderate sampling.
4. Including affected sib pairs, optimal sampling and imprecise ascertainment schemes.

Table 39 The most powerful statistics for various sibling pair samples when model parameters are unknown

	Type of samples from mixture-of-normals model						
BEST STATISTICS	Population sampling	Single-proband sampling	Multiple-proband sampling				
			EDSP	MDSP	Concordant[3]	EDAC[3]	Non-standard designs[4]
Correlation-based							
XU	+						
HE-COM-correlation		+					
IBD-sharing							
IBD-empirical.variance[1]			+				
Combination							
HE-COM-combination							
SCORE3							
RDP			+	+			
Composite[2]			+	+	+	+	

Notes are the same as for table 38.

To answer the second question of what kinds of sampling are advantageous in what situations, we compared different sampling designs using the best statistic for each, which were identified in step one. We found that some design questions are more amenable to statistical investigation, while others should be motivated by scientific and practical considerations, not statistical. We showed that a "sensible" EDAC design means one in which we sample significantly more discordant pairs than concordant pairs. We concluded that in the situations where no phenotyping/enrollment has been done ahead of time, a sensible EDAC design is most powerful as well as most cost effective when genotyping numbers must be low. In addition, we also noted that the use of combination statistics has given the traditional EDAC design more flexibility by allowing moderate sampling strategy and by lessening the need for searching a perfect EDAC sample use exactly the right ratio of discordant to concordant pairs. When single probands have already been phenotyped, it is most efficient to use a single-proband design, unless it is desirable to get genotyping numbers as low as possible. If a sample of pairs has already been phenotyped, clearly optimal sampling is the best choice. Finally, if the number of families available for screening is the primary limitation, single proband sampling that phenotypes both siblings is recommended.

Another accomplishment of this dissertation is the contribution to the theory of QTL mapping methods. We developed robust variants of score statistics for selected sibling pairs. We worked out the theory of combination statistics that is helpful in understanding the relationship among various QTL mapping statistics, and in suggesting new robust and powerful methods for selected samples, such as the RDP statistic. We emphasized the important interplay between statistical methods and sampling designs, and we advocated comparing different sampling designs using the best statistic for each as it would be done in real use! Moreover, we pointed

out that the availability of combination statistics allows for many entirely new experimental designs.

9.2. DISCUSSION

There are some limitations to our studies in the models, in the statistics and in the types of samples we considered. Some open questions are left that deserve further investigation.

9.2.1. Regarding models

The models we considered are limited in that all of our models used an environmental/polygenic sibling correlation of .25. But the size of this covariance will only affect the relative power of ORIGINAL.HE and TRAIT.PRODUCT. In general, the greater the correlation, the better ORIGINAL.HE performs in comparison to TRAIT.PRODUCT. The size of this covariance will also affect the informativeness of the concordant pairs in an EDAC sample. The smaller the correlation, the larger the weights associated with concordant pairs. However, we found that even under a model with correlation of 0, the weights we implemented for our composite statistics were correct.

The performance of the statistics for the non-Gaussian models indicates a need for further exploration. Our non-normal models 1' and 2' are generated by ad-hoc transformation. In all of our studies with different type of sibling pair data, we found that our results for models 1' and 2' were significantly different from those for mixture-of-normals models. Our basic observation is that the most non-parametric statistics performed best against non-normal model, but with RDP statistics, the results on non-normal models were inconclusive (see chapter 6). In addition, Chen et al. (submitted) recently proposed two new score statistics derived from a GEE approach (Chen

et al. 2004) incorporating information on high moments of the trait distribution. It was shown that those high-moment score statistics had highest power with population samples of sibling pairs against a $\chi^2(1)$ distribution, but their behavior on selected samples was unclear. All these facts above suggest further study for non-Gaussian models. Perhaps the most important open questions about trait model are what types of non-Gaussian trait models are most realistic and important, and how various features of the models and statistics interact to determine which statistic is most powerful.

9.2.2. Regarding statistics

There are three combination statistics in the literature that we did not include in our study due to computational limitations. In theory they all should perform very similarly to statistic SCORE3 for sibling pair data. The first statistic we did not include is the ascertainment-corrected variance components statistic proposed by Sham et al. (2000), which conditions on trait values. The second one is the robust regression statistic proposed by Sham et al. (2002), which regresses IBD on trait values. When the variance of π is estimated empirically, this statistic takes exactly the same form as SCORE3 for sibling pairs. Finally, one newest procedure we did not include in our study is the generalized estimating equation (GEE) methods proposed by Chen et al. (2004). They showed that a robust score statistic derived from this GEE approach is equivalent to the regression-based test of Sham et al. (2002). Note that the last three methods mentioned above can all be used for extended pedigrees, while SCORE3 and HE-COM-combination are for sibling pairs,

There are a couple of open questions regarding statistics. As mentioned, parameter misspecification is an important issue in selected samples, especially for those statistics that use population values of trait model parameters. In our sensitivity study, we found, interestingly, that some "wrong" values of the trait mean resulted in higher power than that when correct means

were used. It is not clear how to choose an "optimal" value of the mean to achieve a statistic's maximum power. A few variations on the statistics can be potentially useful but we have not yet considered. For example, we could consider a score statistic that is maximized over the value of trait mean, or even maximized over all parameter values. Similarly, we could construct a statistic that is based on XU (Xu et al. 2000), but with the weights maximized for the particular dataset. However, either of these approaches would entail the loss of a degree of freedom to properly adjust for the maximization. We suspect that as a result there would not be a useful power gain, but a maximized score statistic may be helpful in cases where information about population trait parameters is very poor. Further study is warranted.

9.2.3. Regarding sampling issues

One limitation in the types of single-proband samples we considered is that we studied one-tailed sampling (one sib in the top 10%), ignoring two-tailed sampling (one sib in the top 10% ***or*** in the bottom 10%).We expect that the statistics with incorrect type I error for one-tailed sampling will likely be incorrect also for two-tailed sampling. However, as far as statistical power is concerned, the picture is likely to be altered; because the two-tailed sampling will include all EDAC pairs. We expect combination statistic SCORE3 will be the only powerful statistic with known parameters. Another limitation is that we only considered large sample sizes (large numbers of sibling pairs). We assume that studies with small samples are fairly unusual.

There are also limitations in our study of comparing different sampling designs. We used quantitative approaches and considered simplified situations that can be modeled statistically. However, design choices are *not* purely (or perhaps even predominantly) statistical, and to study only the mathematics is inadequate. Clearly, for any given study, there will be a whole host of practical considerations that are equally important, such as differences in

recruitment effort needed for extreme and non-extreme people, fixed limits on the number of people that can be screened, costs to set up additional study sites, etc. From a mathematical optimization point of view, one could describe this as a situation where every study has both its own cost functions for genotyping and phenotyping (which may not be smooth functions), and also its own constraints. Nevertheless, a vital question is if it is feasible to specify such an "optimal" design in reality, and if such an effort is absolutely necessary.

In terms of the types of samples, the most important limitation is that we only considered sibling pairs. Real studies generally include larger sibships as well (table 1). There is also plenty of evidence that larger sibships are more powerful than sibling pairs, even in the context of discordant designs. For example, Alcais and Abel (2000) and Tang and Siegmund (2001) both showed that if one has ascertained a discordant sibling pair, it is most efficient to also use any other siblings in the sibship - more efficient than recruiting another independent discordant pair. We recommend comprehensive future study of designs that use larger sibships of various types. The most important questions to be explored are likely to be (1) what statistics can be extended to sibships data and how they can be adapted to selected samples, (2) when it is a good idea to recruit the whole sibships instead of collecting the extreme pairs only.

The design comparison of larger sibships samples will be very different from that using sibling pairs only, and will require an entirely new study. For example, one of our results was that discordant-pair only designs are not a good idea. But with larger sibships it might be sufficient to ascertain sibships containing a discordant pair, since some concordant siblings would end up being included as well. On the statistical method issues, our results should be very relevant to larger sibships. For multiple-ascertainment of larger sibships, combination statistics will be essential as they are for sibling pairs. Of the statistics we considered, the score statistic extends naturally to

larger sibships and it is likely to be the best choice. The methods proposed by Sham et al. (2002) and Chen et al. (2004) were developed particularly for extended pedigrees, and they are also good candidates. The composite statistics may also be useful for multiple-ascertainment of larger sibships. It is possible that the different methods for handling larger sibships, particularly selected samples, result in substantial power differences, so further study of these methods is very important. In particular, some additional problems arise with the application of score statistics. One is to figure out what parameter estimates and what denominators to use for selected samples of larger sibships to ensure correct type I error of the statistics. The other difficulty is how to form score statistics to efficiently combine sibships of different sizes over a wide variety of sampling schemes.

APPENDIX A

INDEPENDENCE OF CORRELATION-BASED STATISTICS AND IBD-SHARING STATISTICS

Forrest and Feingold (2000) attempted to give a general proof that correlation-based statistics and IBD-sharing statistics are independent under both the null and alternative hypotheses. They did this in three steps. First, they demonstrated that if there is perfect IBD-sharing information (an infinitely polymorphic marker), the likelihood of the data factors into two pieces, one dependent on the IBD-sharing parameter and the other dependent on the trait parameters. This clearly and simply demonstrates asymptotic independence of the maximum likelihood estimates of the IBD-sharing and correlation parameters. The second step in their proof was to use an expectation-maximization (EM) framework to show that this factorization still holds if the IBD information is estimated from marker data. That argument erroneously defined the observed-data log likelihood (equation 2 of Forrest and Feingold 2000). The third step was to show that both the Haseman-Elston and the maximum likelihood variance components statistics can be approximately derived from the trait-parameter factor of the likelihood.

Overall, the EM formulation made this proof unnecessarily complex. A more straightforward way to look at the issue is simply to note that 1) the likelihood factors when there is perfect IBD information, 2) when we estimate IBD information we do not use trait values in any way, so 3) when we use estimated IBD information we still have independence. Moreover,

the independence between *specific* IBD-sharing and correlation-based statistics can easily be shown algebraically, as follows.

In simple linear regression, let x_i be the independent variable and y_i be the outcome variable. Using the standard results that both $\sum_{i=1}^{n}(x_i-\bar{x})^2$ and $(x_i-\bar{x})$ are independent of $\bar{x}$, we can show that the following regression quantities are independent of $\bar{x}$: $\hat{\beta}_1$, $\hat{\beta}_0$, $\hat{y}$, SSE, MSE or $\hat{\sigma}^2_{Y/x}$. Then the t-statistic for testing the regression coefficient is independent of $\bar{x}$ and this is true under both null and alternative hypothesis, since

$$T = \frac{\hat{\beta}_1}{\sqrt{\hat{\sigma}^2_{\hat{\beta}_1}}} = \frac{\dfrac{\sum_{i=1}^{n}(x_i-\bar{x})y_i}{\sum_{i=1}^{n}(x_i-\bar{x})^2}}{\sqrt{\dfrac{\hat{\sigma}^2_{Y/x}}{\sum_{i=1}^{n}(x_i-\bar{x})^2}}} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})y_i}{\sqrt{\hat{\sigma}^2_{Y/x}}\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}}.$$

The above result can be applied to any of the QTL mapping statistics that are based on linear regression, including the original Haseman-Elston (1972) and HE-COM-correlation statistics that we consider in this dissertation. To do so, we write $x_i = \pi_i - 1/2$, $\bar{x} = \bar{\pi} - 1/2$, and $y_i = A_i$ where A_i can be any function of sibling trait values. Then the t-statistic for testing linkage,

$$T = \frac{\sum A_i(\pi_i - \bar{\pi})}{\sqrt{\hat{\sigma}^2_{A/\pi}}\sqrt{\sum_{i=1}^{n}(\pi_i-\bar{\pi})^2}},$$

is independent of $\bar{\pi}$, and this holds under both the null and alternative hypotheses. Moreover, it holds for perfectly-observed values of π_i, or for values of π_i that are estimated in any way that does not depend on the trait values.

APPENDIX B

MAXIMIZED COMPOSITE STATISTICS

For any of our composite statistics, it is also possible to use a chi-squared type statistic that essentially estimates the optimal weights from particular datasets. Let *a, b* be the components corresponding to the "RDP composite" statistic or Forrest and Feingold's composite statistic (COMPOSITE1). Let *x, y, u* be the components corresponding to the "3-part composite" statistic. Applying Lagrange Multiplier method, we can show that the squared maximized composite statistics take the following forms:

$$a^2 + b^2 \qquad (1)$$

$$x^2 + y^2 + u^2 \qquad (2)$$

which have chi-square distributions with two and three degree of freedoms respectively, under null hypothesis of no linkage. Next, we constructed one-sided tests on the basis of equations (1) and (2), and the resulting statistics are mixed chi-square variables. Then we conducted simulation to find the empirical null distribution and percentiles associated with each of the variables. Finally, we implemented these two tests and found that the maximized composite (1) is less powerful than our linear composite statistics "RDP composite" or COMPOSITE1. And the maximized composite (2) is less powerful than our linear form statistic – the "3-part composite". Furthermore, for EDAC samples, the two component maximized composite statistic (1) showed better performance than its three component counterpart (2). Taken together, we

argue that, in general, such kind of methodology where weights maximized for the particular dataset would entail the loss of a degree of freedom to properly adjust for the maximization, and, more than likely, would not be a useful power gain, despite the inclusion of more maximized component.

APPENDIX C

DATA SIMULATION C++ PROGRAM

```
/*********************************************************************/
/*                      Newsimsib5.c                                 */
/*                                                                   */
/*                                                                   */
/* This program simulates sibling pair data based on user-chosen     */
/* ascertainment criterion, including quantitative trait values under*/
/* any mix-of-normals model and IBD information using a single marker*/
/*                                                                   */
/* It computes the required screening, phenotyping and genotyping    */
/* numbers assuming the situation where no individual has been       */
/* recruited or phenotyped before the study begins.                  */
/*                                                                   */
/* If desired, this program can calculate the average NCP per sibling*/
/* pair based on the formula of Sham and Purcell(2001). In that case,*/
/* population value of some trait model parameters must be entered.  */
/* IMPORTANT: this feature is not well tested and it is not clear if */
/* the formula implemented is a good choice. For maximum accuracy,   */
/* statistical power of any sample should be evaluated using simula- */
/* tion!!!                                                           */
/*                                                                   */
/*                                                                   */
/*                                                                   */
/* A)Compiling                                                       */
/*   Type "g++ Newsimsib5.c -o Newsimsib5.out" to compile and feel   */
/* free to change the name of the executable (Newsimsib5.out).       */
/* IMPORTANT: YOU MUST INCLUDE "IBD.c","simplex.h" and "radom.h" in  */
/* the same directory.                                               */
/*   Description of the simulation program and three associated      */
/* header files:                                                     */
/*                                                                   */
/* 1. "Newsimsib5.c" is the main C++ script which performs the simu- */
/*     lation.                                                       */
/*                                                                   */
/* 2. The file "IBD.c" is a utility which takes genotype information */
/*    at a single marker for two parents and two children and returns*/
/*    the estimated IBD sharing of the two children.                 */
/*                                                                   */
/* 3. The file "random.h" is a utility file which includes several   */
```

```
/*    random number generating programs in C,including those for the */
/*    multinomial and normal/Gaussian distributions.                 */
/*                                                                   */
/* 4.  The file "simplex.h" has a minimization routine taken from    */
/*    "Numerical Recipes in C".                                      */
/*                                                                   */
/*                                                                   */
/*                                                                   */
/* B)ascertainment schemes                                           */
/*                                                                   */
/*    The program allows users to choose an ascertainment scheme     */
/* based on threshold selection.                                     */
/*    Users will be asked to specify their desired selection         */
/* percentiles, e.g., top 10 (x) percent and bottom 10 (y) percent   */
/* for selecting discordant pairs.                                   */
/*    The program generates an empirical distribution of trait values*/
/* under any mixed-of-normals model and computes the actual cutoff   */
/* values of the trait that go with the xth and yth percentiles.     */
/*                                                                   */
/* Note that for fuzzy concordant, the empirical cutoffs are based on*/
/* the distribution of trait-plus-noise and ascertain pairs for whom */
/* trait-plus-noise is over the cutoff for both people.              */
/*                                                                   */
/* Types of ascertainment scheme:                                    */
/*                                                                   */
/* 0)population sample - A random sample, no selection at all.       */
/*                                                                   */
/* 1)single.proband-both.sibs (top tail) - Phenotype both sibs in    */
/*   each pairs screened. Select all sib pairs where at least        */
/*   one sib is above the xth percentile.                            */
/* 2)single.proband-both.sibs (bottom tail) - Phenotype both sibs in */
/*   each pairs screened. Select all sib pairs where at least        */
/*   one sib is below the yth percentile.                            */
/* 3)single.proband-both.sibs (two-tailed) - Phenotype both sibs in  */
/*   each pairs screened. Select all sib pairs where at least one sib*/
/*   is below the yth percentile or above the xth percentile.        */
/*                                                                   */
/* 4)discordant pairs  - Select all sib pairs where one sib is above */
/*   the xth percentile and the other is below the yth percentile.   */
/*                                                                   */
/* 5)high concordant pairs - Select all sib pairs where both sibs    */
/*   are above the xth percentile                                    */
/* 6)low concordant pairs  - Select all sib pairs where both sibs    */
/*    are below the yth percentile.                                  */
/* 7)low + high concordant - Include case (5) and (6)                */
/*                                                                   */
/* 8) EDAC-3.corners (top) - discordant(4) + high concordant(5)      */
/* 9) EDAC-3.corners (bottom) - discordant(4) + low concordant(6)    */
/* 10)EDAC-4.corners - discordant(5) + high and low concordant(7)    */
/*                                                                   */
/* 11)fuzzy concordant high  -  the values of trait-plus-noise for   */
/*    both sibs are above the xth percentile                         */
/* 12)fuzzy concordant low  -  the values of trait-plus-noise for    */
/*    both sibs are below the yth percentile                         */
/*                                                                   */
/* 13)single.proband-one.sib (top tail) - Phenotype only the first   */
/*    sib in each pairs screened. Select all sib pairs where that 1st*/
```

```
/*    sib is above the xth percentile.                                  */
/* 14)single.proband-one.sib (bottom tail) - Phenotype only the first*/
/*    sib in each pairs screened. Select all sib pairs where that 1st*/
/*    sib is below the yth percentile.                                  */
/* 15)single.proband-one.sib (two-tailed) - Phenotype only the first */
/*    sib in each pairs screened. Select all sib pairs where that 1st*/
/*    sib is below the yth percentile or above the xth percentile.   */
/*                                                                      */
/*                                                                      */
/*                                                                      */
/* C)Primary variables:                                               */
/*                                                                      */
/* a. 'p' = QTL biallelic frequency                                   */
/*    'theta'= recombination fraction                                 */
/*    'markerCardinality' is the number of different marker allele   */
/*     versions.                                                       */
/*    "mu_DD", "mu_Dd", and "mu_dd" are the three trait means for    */
/*     the gentoypes DD, Dd, dd where D is for the disease allele.   */
/*    "sd_DD", "sd_Dd", and "sd_dd" are the three trait standard     */
/*     deviations for the gentoypes DD, Dd, dd.                       */
/*    "environmental_cor" is the within-family sibling pair shared   */
/*     environmental correlation.                                     */
/*                                                                    */
/*    "noise_mu" and "noise_sd" are the parameters for the           */
/* normally distributed noise value. They are initialized to be zero */
/* and will remain to be zero for the ascertainment schemes          */
/*  0 through 10, and 13-15. The noise values are non-zero only for  */
/* the fuzzy concordant ascertainment schemes (11 and 12) when users */
/* input non-zero values to noise parameters to simulate a normally  */
/* distributed noise value for each person.                          */
/*                                                                    */
/*                                                                    */
/* b. 'n' is the number of sib pairs to be simulated in a single run */
/*     of the program.                                                */
/*   'Number_Fam_Screened' is the number of families (pairs) screened*/
/*   'Number_Sibs_Phenotyped' is the number of individuals phenotyped*/
/*    Note that the number of people genotyped will always equal to  */
/* (2*n)                                                              */
/*                                                                    */
/*                                                                    */
/*    Note for the combined discordant and concordant sib pair sample*/
/* (or the ascertainment schemes 8,9,10), n_DS, n_CS in the simulated*/
/* sample are calculated.   [n = n_DS + n_CS]                        */
/*                                                                    */
/* c.  'N' is the number of points used to simulate the empirical    */
/*     distribution of trait values from the given population, which */
/*     is  specified by the mix-of-normals trait parameters you use. */
/*     Set N = 100,000 in the current program. N can be change if you*/
/*     wish, with adjustment on the ways to pick up cutoffs.        */
/*                                                                    */
/* d. "type_ascert" is the type of ascertainment you choose.         */
/*                                                                    */
/* For all the ascertainment schemes except 8,9,10:                  */
/*   'y' = the yth percentile of lower_tail.                         */
/*   'x' = the xth percentile of upper_tail.                         */
/*   "low_tail" and "high_tail" are given by the program from the    */
/*   empirical distribution,according to the yth and xth percentiles */
```

```
/*   you specify.  "low_tail" and "high_tail" are, respectively, how */
/*   low and high the two sib traits must be to be included. These   */
/*   define the level of concordance/discordance in the sib pair.    */
/*                                                                   */
/*                                                                   */
/* For ascertainment schemes 8,9,10:                                 */
/* 'y','x','low_tail','high_tail' will be used to select discordant  */
/* pairs; 'y2','x2','low_tail2','high_tail2' will be used to select  */
/* concordant pairs.                                                 */
/*                                                                   */
/*                                                                   */
/* f. An example for MDSP (moderately discordant sibling pairs       */
/*   design: Input x=35, y=35 for upper 35% and lower 35% of         */
/*   the trait distribution.                                         */
/*                                                                   */
/* g. average_NCP is the average NCP per sibling pair based on the   */
/*    NCP formula of Sham and Purcell (2001).                        */
/*                                                                   */
/*                                                                   */
/* D)Subroutines:                                                    */
/*                                                                   */
/* trait_null() returns a single simulated trait value drawn from the*/
/* population.                                                       */
/* quicksort() uses quicksort algorithm.                             */
/* partation() is called by quicksort().                             */
/* select() returns a judgment whether or not the simulated trait    */
/* values of each sib pair screened meet your selection criteria.    */
/* It also returns a value that indicates phenotyping status for each*/
/* pair screened.                                                    */
/*                                                                   */
/* Children() simulates meiosis for sibs based on the parent genotype*/
/*                                                                   */
/* IBD(), runif(),rmult_value() are from the associated header files.*/
/* runif() returns a single Uniform(0,1).                           */
/* rmult_value() is used in the main program to simulate genotype for*/
/* parents.                                                          */
/* IBD() calculates IBD probability for marker alleles.              */
/*                                                                   */
/*                                                                   */
/*                                                                   */
/* E)Input:                                                          */
/*                                                                   */
/* Each time you run the simulation program, you may choose to input */
/* a new set of parameters or to take advantage of existing file     */
/* "simulation_parameters" named by default.                         */
/* Each time when you use a new set of parameters, you will be       */
/* instructed to input all the parameters.                           */
/* The program will generate a file "parameters" for future simula-  */
/* tion using the same set of parameters.                            */
/* You may find it easier to edit the file "parameters" directly when*/
/* you only make minor changes of parameters. Be sure not to change  */
/* any format of the file!!                                          */
/*                                                                   */
/* The file "parameters" contains:                                   */
/*    line 1:  QTL gene frequency                                    */
/*    line 2:  Recombination fraction                                */
/*    line 3:  number of marker allele                               */
```

```
/*    line 4:  population means for DD/Dd/dd                          */
/*    line 5:  population standard deviations for DD/Dd/dd            */
/*    line 6:  environmental correlation                              */
/*    line 7:  type of ascertainment                                  */
/*    line 8:  yth for lower_tail &  xth for upper_tail               */
/*    line 8-2: y2th for lower_tail2 & x2th for upper_tail2           */
/*    line 9:  noise_mu, noise_standard_deviation                     */
/*    line 10:  empirical cuttoffs for low_tail &  upper_tail         */
/*    line 10-2:  empirical cutoffs for low_tail2 &  upper_tail2      */
/*    line 11: the number of replicates                               */
/* Note:                                                              */
/*    a) Multiple numbers within each line are separated by spaces.   */
/*    b) Line 8-2 will appear only for ascertainment 8,9,10.          */
/*                                                                    */
/*                                                                    */
/*                                                                    */
/* F)Output:                                                          */
/*                                                                    */
/* The simulation program generates three output files:               */
/*"simulation_parameters", "sib_simulate","noise_simulate" by default*/
/*                                                                    */
/*                                                                    */
/* The "simulation_parameters" stores the parameters you input from   */
/* the screen and has been described above.                           */
/*                                                                    */
/* The primary output file "sib_simulate" contains 3 parts in order: */
/*   1. trait values of sibling pairs and IBD sharing probabilities   */
/*      (probabilities of sharing 0,1,2 alleles.)                     */
/*   2. "-999" immediately followed in a new line, which serves as    */
/*      the signal of the end of part 1 of the output file.           */
/*   3. Records of the simulation parameters that have been used to   */
/*      generate trait values in part 1; records of sample sizes;     */
/*      and records for NCP calculation if that is applied.           */
/*                                                                    */
/* To see the last two parts of output file, TYPE:                    */
/*  tail -25l sib_simulate                                            */
/*                                                                    */
/*  The other output file "noise_simulate" is very similar to         */
/* "sib_simulate" in the three part structure but:                    */
/*  In the part 1 of "noise_simulate":                                */
/*     Columns 1-2: noise values for each pair                        */
/*     Columns 3-4: QTL trait values for each pairs                   */
/*     Columns 5-6: the trait-plus-noise values for the pairs.        */
/*                                                                    */
/*                                                                    */
/*                                                                    */
/* G) "sib_simulate" is the file to be read in by the statistical     */
/*    program                                                         */
/********************************************************************/



// TYPE "g++ Newsimsib5.c" TO COMPLILE
// MUST INCLUDE "IBD.c","simplex.h" and "radom.h" in the same directory
// MUST INCLUDE FILE "parameters" IN THE SAME DIRECORY, IF NOT TO CREAT
// A NEW ONE
```

```
#include<stdlib.h>
#include<stdio.h>
#include<math.h>
#include<limits.h>
#include<time.h>
#include<iostream.h>
#include "random.h"
#include "IBD.c"
#include "simplex.h"
#include<assert.h>
#include<iomanip.h>


typedef double Item;
#define key(A) (A)
#define less(A, B) (key(A) < key(B))
#define exch(A, B) { Item t = A; A = B; B = t; }


//GLOBAL VARIABLES ARE DEFINED HERE FOR CONVENIENCE
float p;  /*QTL frequency*/
float theta = 0.00 ; /*recombination fraction */
int markerCardinality; /*number of marker allele*/
float mu_DD, mu_Dd, mu_dd;  /* trait means for each genotype*/
float sd_DD, sd_Dd, sd_dd;  /* standard deviation for each genotype*/
float environmental_cor;    /* trait model, environmental correlation*/
float noise_mu=0.0, noise_sd=0.0;   /* noise mean and sd */




main()
{

  ////////////////////////////////////
  //VARIABLE AND FUNTION DECLARATIONS
  ////////////////////////////////////
  void ChildGen(long *, long *, long *, long *, long *, long *, double);
  void quicksort(Item a[], int l, int r);
  Item trait_null( );
  int select(Item a[],int choice, double trait_sib1, double trait_sib2,
float low, float high, float low2, float high2, int *, int *, int *, int*);

  long m = 0, j = 0, count = 0;

  /* cutoff value for the trait marginal distribution*/
   float low_tail, high_tail;
   float low_tail2, high_tail2;  /* for ascertainment 8-10, concordance */

  int n; /* number of replicates*/
  int n_DS = 0, n_CS = 0 ;    /* for ascertainment 8-10 n=n_DS+n_CS */
  int n_CSL = 0, n_CSH =0 ;   /* n_CS = n_CSL + n_CSH */
  int *p1 = & n_DS;   /* pointers for passing values from select() */
  int *p2 = & n_CSL;
  int *p3 = & n_CSH;
```

```
  int type_ascert;       /* type of ascertainment */
  int y=99,x=99; /* yth & xth percentile*/
  int y2=99,x2=99; /* for ascertainment 8-10, concordance */

  int c = 0; int c_NCP = 0;
  int True = 0; /* indicator for case_status*/
  int phenotyping_status = 2; /*may be changed to 1 in subroutine select()*/
  long Number_Fam_Screened = 0; /*the number of screened families (pairs)*/
  long Number_Sibs_Phenotyped = 0; /* # of people actually phenotyped */

  int i, N=100000;  /*100000 points simulated for the empirical
distribution*/
  Item a[N];


  //IO,"parameters" & "sib_simulate"
  FILE *sib_outfile;             /*poiner for output file "sib_simulate"*/
  sib_outfile = fopen ("sib_simulate","w");
  FILE *noise_outfile;           /*poiner for output file "noise_simulate"*/
  noise_outfile = fopen ("noise_simulate","w");
  FILE *parameter_file;   /*pointer for input file "simulation_parameters"*/


  ////////////////////////////////////////////////////////////////
  //CHOOSE TO USE A NEW SET OF PARAMETERS OR TO USE EXISTING INPUT FILE
  //THAT IS CALLED 'simulation_parameters' BY DEFAULT
  ////////////////////////////////////////////////////////////////
  printf("Create new parameter file? (y/n)\n");
  c = getchar();

  //IF "YES",INPUT A NEW SET OF PARAMETERS AND CREATE A FILE "parameters"
if (c=='y'|| c=='Y') {
   parameter_file = fopen ("simulation_parameters","w");
   printf ("\nPlease input parameters for mixture-of-normals model \n");
   printf ("\nQTL gene frequency:\n");
   scanf("%f", &p);
   fprintf(parameter_file, "\n%6.2f\n",p) ;

   printf ("Recombination fraction:\n");
   scanf("%f", &theta);
   fprintf(parameter_file, "%6.2f\n",theta);

   printf ("Number of (single) marker allele:\n");
   scanf("%d", &markerCardinality);
   fprintf(parameter_file,"%6d\n ",markerCardinality);

   printf("\nTrait means (normal model) for genotypes DD/Dd/dd where D is for
the QTL\n");
   printf ("mu_DD:\n");
   scanf("%f", &mu_DD);
   printf ("mu_Dd:\n");
   scanf("%f", &mu_Dd);
   printf ("mu_dd:\n");
   scanf("%f", &mu_dd);
   fprintf(parameter_file,"%6.2f %6.2f %6.2f\n",mu_DD,mu_Dd,mu_dd);
```

```
   printf("\nTrait standard deviations (normal model) for genotype
DD/Dd/dd\n");
   printf("sd_DD:\n");
   scanf("%f", &sd_DD);
   printf ("sd_Dd:\n");
   scanf("%f", &sd_Dd);
   printf ("sd_dd:\n");
   scanf("%f", &sd_dd);
   fprintf(parameter_file,"%6.3f %6.3f %6.3f\n", sd_DD,sd_Dd,sd_dd);

  printf ("\nwithin-family sibling shared environmental correlation:\n");
  scanf("%f",&environmental_cor);
  fprintf(parameter_file, "%6.2f\n",environmental_cor);

  printf ("\ncut_off values for the trait marginal distribution\n");
  printf("\nTypes of ascertainment:\n\n");

printf("0)  non-ascertained population sample\n\n");
printf("1)  single.proband-both.sibs (top tail)\n");
printf("    - Phenotype both sibs in each pair screened and select all \n
sib pairs where at least one sib is above the xth percentile.\n");
printf("2)  single.proband-both.sibs (bottom tail)\n");
printf("    - Phenotype both sibs in each pair screened and select all \n
sib pairs where at least one sib is below the yth percentile.\n");
printf("3)  single.proband-both.sibs (two-tailed)\n");
printf("    - Phenotype both sibs in each pair screened and select all \n
sib pairs where at least one sib is below the yth percentile \n       or
above the xth percentile.\n\n");
printf("4)  discordant pairs\n");
printf("    - Select all sib pairs where one sib is above \n       the xth
percentile and the other is below the yth percentile.\n\n");
printf("5)  high concordant pairs\n");
printf("    - Select all sib pairs where both sibs are above the xth
percentile.\n");
printf("6)  low concordant pairs\n");
printf("    - Select all sib pairs where both sibs are below the yth
percentile.\n");
printf("7)  low + high concordant\n\n");
printf("8)  EDAC-3.corners-top: discordant + high concordant\n");
printf("9)  EDAC-3.corners-bottom: discordant + low concordant\n");
printf("10) EDAC-4.corners: discordant + high and low concordant\n\n");
printf("11) fuzzy concordant high\n");
printf("    - Select all sib pairs where the values of \n       trait-plus-
noise for both sibs are above the xth percentile.\n");
printf("12) fuzzy concordant low\n");
printf("    - Select all sib pairs where the values of \n       trait-plus-
noise for both sibs are below the yth percentile.\n\n");
printf("13)  single.proband-one.sib (top tail)\n");
printf("    - Phenotype only the first sib in each pair screened and \n
select all sib pairs where that 1st sib is above the xth percentile.\n");
printf("14)  single.proband-one.sib (bottom tail)\n");
printf("    - Phenotype only the first sib in each pair screened and \n
select all sib pairs where that 1st sib  is below the yth percentile.\n");
printf("15)  single.proband-one.sib (two-tailed)\n");
printf("    - Phenotype only the first sib in each pair screened and \n
select all sib pairs where that 1st sib is below the yth percentile  \n
or above the xth percentile.\n\n");
```

```
scanf("%d",&type_ascert);
fprintf(parameter_file, "%6d\n",type_ascert);

 if  ( type_ascert == 0 ) {
   printf( "\nYou have chosen non-ascertained population sample\n");
   fprintf(parameter_file, "%6d  %6d\n",99,99);

 }else if  (type_ascert ==1) {
   printf("Please enter the desired xth percentile of upper_tail\n");
   scanf("%d",&x);
   fprintf(parameter_file, "%6d  %6d\n",99,x);
   printf("y %d x %d\n",y,x);

 }else if  (type_ascert ==2) {
   printf("Please enter the desired yth percentile of lower_tail\n");
   scanf("%d",&y);
   fprintf(parameter_file, "%6d  %6d\n",y,99);
   printf("y %d x %d\n",y,x);

 }else if  (type_ascert ==13) {
   printf("Please enter the desired xth percentile of upper_tail\n");
   scanf("%d",&x);
   fprintf(parameter_file, "%6d  %6d\n",99,x);
   printf("y %d x %d\n",y,x);

 }else if  (type_ascert ==14) {
   printf("Please enter the desired yth percentile of lower_tail\n");
   scanf("%d",&y);
   fprintf(parameter_file, "%6d  %6d\n",y,99);
   printf("y %d x %d\n",y,x);

}else if  (type_ascert ==5) {
   printf("Please enter the desired xth percentile of upper_tail\n");
   scanf("%d",&x);
   fprintf(parameter_file, "%6d  %6d\n",99,x);
   printf("y %d x %d\n",y,x);

 }else if  (type_ascert ==6) {
   printf("Please enter the desired yth percentile of lower_tail\n");
   scanf("%d",&y);
   fprintf(parameter_file, "%6d  %6d\n",y,99);
   printf("y %d x %d\n",y,x);

 }else if  (type_ascert ==11 ) {
   printf("\nNoise Model parameter - mean\n");
   printf ("noise_mu:\n");
   scanf("%f", &noise_mu);
   printf("\nNoise Model parameter - standard deviation\n");
   printf("noise_sd:\n");
   scanf("%f", &noise_sd);
   printf("\nPlease enter the desired xth percentile of upper_tail\n");
   scanf("%d",&x);
   fprintf(parameter_file, "%6d  %6d\n",99,x);
   printf("y %d x %d\n",y,x);

 }else if  (type_ascert ==12 ) {
```

```
   printf("\nNoise Model parameter - mean\n");
   printf ("noise_mu:\n");
   scanf("%f", &noise_mu);
   printf("\nNoise Model parameter - standard deviation\n");
   printf("noise_sd:\n");
   scanf("%f", &noise_sd);
   printf("\nPlease enter the desired yth percentile of lower_tail\n");
   scanf("%d",&y);
   fprintf(parameter_file, "%6d  %6d\n",y,99);
   printf("y %d x %d\n",y,x);

 }else if (type_ascert == 3 || type_ascert == 15 || type_ascert == 4 ||
type_ascert == 7 ) {
   printf("Please enter the desired xth percentile of upper_tail\n");
   scanf("%d",&x);
   printf("Please enter the desired yth percentile of lower_tail\n");
   scanf("%d",&y);
   fprintf(parameter_file, "%6d  %6d\n",y,x);
   printf("y %d x %d\n",y,x);

 }else if (type_ascert == 8) {
   /* variables x, y are for discordance */
   puts(" Please enter the criteria to select discordant pairs first:\n");
   printf("Please enter the desired xth percentile of upper_tail for
discordance\n");
   scanf("%d",&x);
   printf("Please enter the desired yth percentile of lower_tail for
discordance\n");
   scanf("%d",&y);
   fprintf(parameter_file, "%6d  %6d\n",y,x);
   printf("y %d x %d\n",y,x);
   /* variables x2, y2 are for concordance */
   puts(" \nPlease enter the criteria to select concordant pairs:\n");
   printf("Please enter the desired x2th percentile of upper_tail for
concordance\n");
   scanf("%d",&x2);
   fprintf(parameter_file, "%6d  %6d\n",99,x2);
   printf("y2 %d x2 %d\n",y2,x2);

 }else if (type_ascert == 9) {
   /* variables x, y are for discordance */
   puts(" Please enter the criteria to select discordant pairs first:\n");
   printf("Please enter the desired xth percentile of upper_tail for
discordance\n");
   scanf("%d",&x);
   printf("Please enter the desired yth percentile of lower_tail for
discordance\n");
   scanf("%d",&y);
   fprintf(parameter_file, "%6d  %6d\n",y,x);
   printf("y %d x %d\n",y,x);
   /* variables x2, y2 are for concordance */
   puts("\nPlease enter the criteria to select concordant pairs:\n");
   printf("Please enter the desired y2th percentile of lower_tail\n");
   scanf("%d",&y2);
   fprintf(parameter_file, "%6d  %6d\n",y2,99);
   printf("y2 %d x2 %d\n",y2,x2);
```

```
 }else if (type_ascert == 10) {
   /* variables x, y are for discordance */
   puts(" Please enter the criteria to select discordant pairs first:\n");
   printf("Please enter the desired xth percentile of upper_tail for
discordance\n");
   scanf("%d",&x);
   printf("Please enter the desired yth percentile of lower_tail for
discordance\n");
   scanf("%d",&y);
   fprintf(parameter_file, "%6d  %6d\n",y,x);
   printf("y %d x %d\n",y,x);
   /* variables x2, y2 are for concordance */
   puts("\nPlease enter the criteria to select concordant pairs:\n");
   printf("Please enter the desired x2th percentile of upper_tail for
concordance\n");
   scanf("%d",&x2);
   printf("Please enter the desired y2th percentile of lower_tail for
concordance\n");
   scanf("%d",&y2);
   fprintf(parameter_file, "%6d  %6d\n",y2,x2);
   printf("y2 %d x2 %d\n",y2,x2);

 } //match if

fprintf(parameter_file,"%6.3f %6.3f\n", noise_mu, noise_sd);


////////////////////////////////////////////////////////////////////
//GENERATE EMPIRICAL DISTRIBUTION & FIND THE CUTOFFS; N=10,0000 HERE
////////////////////////////////////////////////////////////////////
        for (i = 0; i < N; i++)
           a[i] = trait_null();
        //for (i = 0; i < N; i++) printf("%6.5f\n ", a[i]);

        quicksort(a,0,N-1);
        //printf("sorted\n");
        //for (i = 0; i < N; i++) printf("%6.5f\n ", a[i]);

      /* Find cuttoff*/
      low_tail= a[1000*y-1];
      high_tail= a[99999-1000*x];
      low_tail2= a[1000*y2-1];
      high_tail2= a[99999-1000*x2];


/*write corresponding empirical cutoff values into "simulation_parameters"*/

     if  ( type_ascert == 0 ) {
       low_tail = a[99999]; high_tail=a[0];
       printf( "\nYou have chosen non-ascertained population sample\n");
       fprintf(parameter_file, "%6.7f  %6.7f\n",a[99999],a[0]);

     }else if  (type_ascert ==1 || type_ascert ==13) {
       low_tail = a[99999];
       printf("\nSingle-proband top tail\n");
       printf("%6.7f\n",high_tail);
```

```
       printf("%4.2f quantile: %6.7f; %4.2f quantile:
%6.7f\n",y/100.0,low_tail,(1-x/100.0),high_tail);
       fprintf(parameter_file, "%6.7f  %6.7f\n",a[99999],high_tail);

     }else if  (type_ascert ==2 || type_ascert ==14) {
       high_tail = a[0];
       printf("\nSingle-proband bottom tail\n");
       printf("%6.7f\n",low_tail);
       printf("%4.2f quantile: %6.7f; %4.2f quantile:
%6.7f\n",y/100.0,low_tail,(1-x/100.0),high_tail);
       fprintf(parameter_file, "%6.7f  %6.7f\n",low_tail,a[0]);

     }else if  (type_ascert ==5 || type_ascert ==11) {
       low_tail = a[99999];
       printf("\nConcordant top tail\n");
       printf("%6.7f\n",high_tail);
       printf("%4.2f quantile: %6.7f; %4.2f quantile: %6.7f\n",(1-
x/100.0),high_tail,(1-x/100.0),high_tail);
       fprintf(parameter_file, "%6.7f  %6.7f\n",a[99999],high_tail);

     }else if  (type_ascert ==6 || type_ascert ==12) {
       high_tail = a[0];
       printf("\nConcordant bottom tail\n");
       printf("%6.7f\n",low_tail);
       printf("%4.2f quantile: %6.7f; %4.2f quantile: %6.7f\n",y/100.0,
low_tail, y/100.0,low_tail);
       fprintf(parameter_file, "%6.7f  %6.7f\n",low_tail,a[0]);

     }else if  ( type_ascert == 3 || type_ascert == 15 || type_ascert == 4 ||
type_ascert == 7) {
       printf("\nTop tail and bottom tail\n");
       printf("%6.7f  %6.7f\n",low_tail,high_tail);
       printf("%4.2f quantile: %6.7f; %4.2f quantile:
%6.7f\n",y/100.0,low_tail,(1-x/100.0),high_tail);
       fprintf(parameter_file, "%6.7f  %6.7f\n",low_tail,high_tail);

     }else if  ( type_ascert == 8) {
       /* cutoff for discordance */
       printf("\nTop tail and bottom tail for discordance\n");
       printf("%6.7f  %6.7f\n",low_tail,high_tail);
       printf("%4.2f quantile: %6.7f; %4.2f quantile:
%6.7f\n",y/100.0,low_tail,(1-x/100.0),high_tail);
       fprintf(parameter_file, "%6.7f  %6.7f\n",low_tail,high_tail);
       /* cutoff for concordance high */
       low_tail2 = a[99999];
       printf("\nConcordant top tail\n");
       printf("%6.7f\n",high_tail2);
       printf("%4.2f quantile: %6.7f; %4.2f quantile: %6.7f\n",(1-
x2/100.0),high_tail2,(1-x2/100.0),high_tail2);
       fprintf(parameter_file, "%6.7f  %6.7f\n",a[99999],high_tail2);

     }else if  ( type_ascert == 9) {
       /* cutoff for discordance */
       printf("\nTop tail and bottom tail for discordance\n");
       printf("%6.7f  %6.7f\n",low_tail,high_tail);
       printf("%4.2f quantile: %6.7f; %4.2f quantile:
%6.7f\n",y/100.0,low_tail,(1-x/100.0),high_tail);
```

```
       fprintf(parameter_file, "%6.7f  %6.7f\n",low_tail,high_tail);
       /* cutoff for concordance low */
       high_tail2 = a[0];
       printf("\nConcordant bottom tail\n");
       printf("%6.7f\n",low_tail2);
       printf("%4.2f quantile: %6.7f; %4.2f quantile:
%6.7f\n",y2/100.0,low_tail2, y2/100.0,low_tail2);
       fprintf(parameter_file, "%6.7f  %6.7f\n",low_tail2,a[0]);

     }else if  ( type_ascert == 10) {
       /* cutoff for discordance */
       printf("\nTop tail and bottom tail for discordance\n");
       printf("%6.7f  %6.7f\n",low_tail,high_tail);
       printf("%4.2f quantile: %6.7f; %4.2f quantile:
%6.7f\n",y/100.0,low_tail,(1-x/100.0),high_tail);
       fprintf(parameter_file, "%6.7f  %6.7f\n",low_tail,high_tail);
       /* cutoff for concordance */
       printf("\nTop tail and bottom tail for concordance\n");
       printf("%6.7f  %6.7f\n",low_tail2, high_tail2);
       printf("%4.2f quantile: %6.7f; %4.2f quantile:
%6.7f\n",y2/100.0,low_tail2,(1-x2/100.0),high_tail2);
       fprintf(parameter_file, "%6.7f  %6.7f\n",low_tail2,high_tail2);

     } //match if



  printf ("\n Please input the number of replicates desired:\n");
  scanf("%d", &n);
  fprintf(parameter_file, "\n%6d\n\n",n) ;
  fclose(parameter_file);


////////////////////////////////////////////////////
//IF "NO", READ IN PARAMETERS FROM EXISTING INPUT FILE
////////////////////////////////////////////////////
 } else if ( c == 'n' || c == 'N') {
              if (( parameter_file = fopen("simulation_parameters","r")) ==
NULL) {
                    fprintf(stderr,"Cannot open file
'simulation_parameters'\n");
                    exit (1);
               }else {
                printf("reading exiting file\n");
                    fscanf(parameter_file,"%f", &p);
                    printf("allele prequency:%f\n",p) ;
                    fscanf(parameter_file,"%f", &theta);
                    printf ("Recombination fraction:%6.2f\n",theta);
                    fscanf(parameter_file,"%d", &markerCardinality);
                    printf ("Number of marker allele:%6d\n
",markerCardinality);
                    fscanf(parameter_file,"%f%f%f\n",&mu_DD,&mu_Dd,&mu_dd);
                    printf("means:%6.2f %6.2f %6.2f\n",mu_DD,mu_Dd,mu_dd);
                    fscanf(parameter_file,"%f %f %f\n",
                           &sd_DD,&sd_Dd,&sd_dd);
                    printf("sd:%6.3f %6.3f %6.3f\n", sd_DD,sd_Dd,sd_dd);
                    fscanf(parameter_file,"%f",&environmental_cor);
```

```
                    printf ("environmental
correlation:%6.2f\n",environmental_cor);
                    fscanf(parameter_file, "%d\n",&type_ascert);
                    fscanf(parameter_file, "%d  %d\n",&y,&x);
                    printf("\ntype of ascertainment is :%6d\n",type_ascert);
                    printf("\nyth percentile of lower_tail & xth percentile
of upper_tail :%6d    %6d  \n",y,x);

        if ( type_ascert == 8 || type_ascert == 9 || type_ascert == 10 ) {
                  fscanf(parameter_file, "%d  %d\n",&y2,&x2);
                  puts("above criteria are for discordance; now the criteria
for concordance:");
                  printf("y2th percentile of lower_tail2 & x2th of
upper_tail2 : %6d %6d  \n",y2,x2);
        } //match if


                    fscanf(parameter_file,"%f%f\n",&noise_mu,&noise_sd);
                    printf("\nnoise_mu:%6.2f,
noise_sd:%6.2f\n",noise_mu,noise_sd);
                    fscanf(parameter_file,"%f %f",&low_tail,&high_tail);
                    printf("\nlow_high_tails:%6.7f
%6.7f\n",low_tail,high_tail);

         if ( type_ascert == 8 || type_ascert == 9 || type_ascert == 10 ) {
                 puts("above criteria are for discordance; now the criteria
for concordance:");
                 fscanf(parameter_file,"%f %f",&low_tail2,&high_tail2);
                 printf("low_high_tails:%6.7f
%6.7f\n",low_tail2,high_tail2);
         } //match if


                    fscanf(parameter_file,"%d", &n);
                    printf("\nThe number of replicates desired:%6d\n\n",n) ;

                    fclose(parameter_file);

         } // match if c=="n"
  } else {
   printf("You must type y/n. \n");
   exit(2);
  }


/* paraemters required by Sham&Purcell(2001)page1529*/
 float pop_mean = 0.0;  /*population trait mean*/
 float pop_std = 0.0;   /*population trait standard deviation*/
 float pop_sibcorr = 0.0; /*population sibling trait correlation*/
 float overall_heritability = 0.0;
/* proportion of phenotypic variance explained by additive effects of the
QTL*/
 double NCP1 = 0.0; /* partial formula for NCP of a sib pair*/
 double average_NCP =0.0; /* average NCP per sibpair*/

 ////////////////////////////////////////
 //Enter parameters for calulating NCP
```

```
 ////////////////////////////////////////
 printf ("\nDo you want to compute NCP_per_sibpair based on
Sham&Purcell(2001)? (y/n)\n");
 c_NCP = getchar();
  c_NCP = getchar();

 if (c_NCP=='y' || c_NCP=='Y') {
 printf ("\nPlease input population trait parameters for NCP calculation
\n");
 printf ("\npopulation trait mean:\n");
 scanf("%f", &pop_mean);
 printf ("\npopulation trait standard deviation:\n");
 scanf("%f", &pop_std);
 printf ("\npopulation sibling trait correlation :\n");
 scanf("%f", &pop_sibcorr);
 printf ("\noverall_heritability:\n");
 scanf("%f", &overall_heritability);
 }


  double trait_sib1=0.0,trait_sib2=0.0;
  double noise_sib1=0.0,noise_sib2=0.0;
  double total_trait_sib1=0.0,total_trait_sib2=0.0;
  double sib1_mean, sib1_sd;
  double QTL_freq[2] = {0.0};
  double IBDprob[3] = {0.0};
  double * markerFrequency = new double[markerCardinality];


  long * matDSLgene      = new long[2],
       * patDSLgene      = new long[2],
       * maternalMarker  = new long[2],
       * paternalMarker  = new long[2],
       * MatChrom1       = new long[2],
       * MatChrom2       = new long[2],
       * PatChrom1       = new long[2],
       * PatChrom2       = new long[2],
       * Child1DSL       = new long[2],
       * Child2DSL       = new long[2],
       * Child1Marker    = new long[2],
       * Child2Marker    = new long[2];

  ////////////////////////////////////////
  //SIMULATE GENOTYPE FOR PARENTS
  ////////////////////////////////////////
  for(m=0 ; m<markerCardinality ; m++)    /*maker freq, equal prob (1/n) */
    markerFrequency[m] = 1.0/((double)markerCardinality);

  QTL_freq[0] = 1.0 - p; /*QTL freq[0]-normal allele*/
  QTL_freq[1] = p;       /*QTL freq[0]-disease allele*/


  matDSLgene[0] = matDSLgene[1] = patDSLgene[0] = patDSLgene[1] = 0;
  maternalMarker[0] = maternalMarker[1] = paternalMarker[0] =
paternalMarker[1]   = 0;
  MatChrom1[0] = MatChrom1[1] = MatChrom2[0] = MatChrom2[1] = 0;
  PatChrom1[0] = PatChrom1[1] = PatChrom2[0] = PatChrom2[1] = 0;
```

```
  Child1DSL[0] = Child1DSL[1] = Child2DSL[0] = Child2DSL[1] = 0;
  Child1Marker[0] = Child1Marker[1] = Child2Marker[0] = Child2Marker[1] = 0;

 long fam = 0;
 while((fam++)<n){
       rmult_value(2,2,QTL_freq,matDSLgene);
       rmult_value(2,2,QTL_freq,patDSLgene);
       rmult_value(2,markerCardinality,markerFrequency,paternalMarker);
       rmult_value(2,markerCardinality,markerFrequency,maternalMarker);

       if(runif() < 0.50 ){
             MatChrom1[0] = maternalMarker[0]; MatChrom1[1] = matDSLgene[0];

             MatChrom2[0] = maternalMarker[1]; MatChrom2[1] = matDSLgene[1];
       }
       else{
             MatChrom1[0] = maternalMarker[0]; MatChrom1[1] = matDSLgene[1];
             MatChrom2[0] = maternalMarker[1]; MatChrom2[1] = matDSLgene[0];
       }

       if(runif() < 0.50 ){
             PatChrom1[0] = paternalMarker[0]; PatChrom1[1] = patDSLgene[0];
             PatChrom2[0] = paternalMarker[1]; PatChrom2[1] = patDSLgene[1];
       }
       else{
             PatChrom1[0] = paternalMarker[0]; PatChrom1[1] = patDSLgene[1];
             PatChrom2[0] = paternalMarker[1]; PatChrom2[1] = patDSLgene[0];
       }


   ////////////////////////////////////////////////
  // MEIOSIS BASED ON PARENTAL GENOTYPES and theta
  // SIMULATE GENOTYPES AND COMPUTE IBD PROBABILITY FOR SIBLING PAIRS
  ////////////////////////////////////////////////////////////////////
      ChildGen(MatChrom1,MatChrom2,PatChrom1,PatChrom2,Child1Marker,
                Child1DSL,theta);

       ChildGen(MatChrom1,MatChrom2,PatChrom1,PatChrom2,Child2Marker,
            Child2DSL,theta);

       IBD(maternalMarker,paternalMarker,Child1Marker,
              Child2Marker,&IBDprob[0]);


       if( IBDprob[0]+IBDprob[1]+IBDprob[2] < 0.999999999)
           {while(1) printf("ERROR in IBD!!!\n");}

//////////////////////////////////////////////////////
// SIMULATE Kids' QTL phenotypes (correlated bivariate normal)
//////////////////////////////////////////////////////
   if( Child1DSL[0] + Child1DSL[1] == 2 ) { trait_sib1 = rnorm()*sd_dd +
mu_dd;
       sib1_mean = mu_dd; sib1_sd = sd_dd;}
   if( Child1DSL[0] + Child1DSL[1] == 3 ) { trait_sib1 = rnorm()*sd_Dd +
mu_Dd;
       sib1_mean = mu_Dd; sib1_sd = sd_Dd;}
```

```
   if( Child1DSL[0] + Child1DSL[1] == 4 ) { trait_sib1 = rnorm()*sd_DD +
mu_DD
;  sib1_mean = mu_DD; sib1_sd = sd_DD;}

   if( Child2DSL[0] + Child2DSL[1] == 2 ) {
       trait_sib2 = (mu_dd + environmental_cor*sd_dd*(trait_sib1-
sib1_mean)/sib1_sd)+
         rnorm()*sd_dd*sqrt(1-environmental_cor*environmental_cor);
       }
   if( Child2DSL[0] + Child2DSL[1] == 3 ) {
       trait_sib2 = (mu_Dd + environmental_cor*sd_Dd*(trait_sib1-
sib1_mean)/sib1_sd)+
         rnorm()*sd_Dd*sqrt(1-environmental_cor*environmental_cor);
       }
   if( Child2DSL[0] + Child2DSL[1] == 4 ) {
       trait_sib2 = (mu_DD + environmental_cor*sd_DD*(trait_sib1-
sib1_mean)/sib1_sd)+
         rnorm()*sd_DD*sqrt(1-environmental_cor*environmental_cor);
       }

////////////////////////////////////////////////////////////////////////
//SIMULATE noise value for each person's i.i.d. normal with parameters
//noise_mu and noise_sd; and use the value of trait_plus_noise to select.
//For ascertainment 0 through 10, noise value remain zero;
//For fussy concordant, noise values are non-zero.
////////////////////////////////////////////////////////////////////////
noise_sib1 = rnorm()*noise_sd + noise_mu;
noise_sib2 = rnorm()*noise_sd + noise_mu;
total_trait_sib1= trait_sib1+ noise_sib1;
total_trait_sib2= trait_sib2+ noise_sib2;

/*testing commands*/
// printf("trait1,  trait2,  noise1,  noise2,  total1,  total2\n");
//
printf("%8.2f,%8.2f,%8.2f,%8.2f,%8.2f,%8.2f\n",trait_sib1,trait_sib2,noise_si
b1,noise_sib2,total_trait_sib1,total_trait_sib2);

////////////////////////////////////////////////////////////////////////
//SELECT SIBS THAT MEET YOUR REQUIRMENTS based on trait or trait-plus-
nonzero-
//noise; store trait and noise values for each selected person separately.
////////////////////////////////////////////////////////////////////////
/
 True =
select(a,type_ascert,total_trait_sib1,total_trait_sib2,low_tail,high_tail,low
_tail2,high_tail2, p1, p2, p3, &phenotyping_status);

 if (True == 1) {

           /* output simulated trait values to file sib_simulate*/
           fprintf(sib_outfile,"%8.4f %8.4f   %6.2f %6.2f %6.2f
\n",trait_sib1, trait_sib2,IBDprob[0],IBDprob[1],IBDprob[2]);
           fprintf(noise_outfile,"%8.4f %8.4f      %8.4f %8.4f      %8.4f
%8.4f\n", noise_sib1, noise_sib2, trait_sib1, trait_sib2, total_trait_sib1,
total_trait_sib2);

         /*Calculate NCP per sib pair by Sham&Purcell(2001)*/
```

```
         if (c_NCP=='y' || c_NCP=='Y') {
           NCP1 = pow(((trait_sib1+trait_sib2-
2*pop_mean)/(pop_std*(1+pop_sibcorr))),2.0)-pow(((trait_sib1-
trait_sib2)/(pop_std*(1-pop_sibcorr))),2.0);
           average_NCP += pow((NCP1+4*pop_sibcorr/(1-
pop_sibcorr*pop_sibcorr)),2.0);}

            Number_Fam_Screened++ ;
            Number_Sibs_Phenotyped += phenotyping_status; /*always plus 2*/

      }else {
           Number_Fam_Screened++;
             Number_Sibs_Phenotyped += phenotyping_status; /* +1 or +2 */
             fam--;   /*do not count sibs that not meet the cut-off*/

      }

 //testing command
 //printf("\nin main: phenotyping_status %d,Number_Fam_Screened %d,
Num_Sibs_Phenotyped %d\n", phenotyping_status, Number_Fam_Screened,
Number_Sibs_Phenotyped);

  }  //matches while loop so that fam=n replicates


printf("\n\nDone!\n\n");
n_CS = n_CSL + n_CSH; /*calculation for ascertainment 8 to 10*/

if (c_NCP=='y' || c_NCP == 'Y') {
   average_NCP =
average_NCP*(overall_heritability*overall_heritability)/(16*n);
   printf("%6.3f   << average_NCP_per_sibpair\n\n",average_NCP);
   printf("%6.3f %6.3f %6.3f %6.3f  << input population parameters
mean/std/corr/H^2\n\n",pop_mean,pop_std,pop_sibcorr,overall_heritability);
}

printf("%6d     << total # of families (pairs) screened\n",
Number_Fam_Screened);
printf("%6d     << total # of families (pairs) ascertained after
examining\n",n) ;
printf("%6d     << total # of people phenotyped \n", Number_Sibs_Phenotyped);
printf("%6d     << total # of people genotyped\n\n", 2*n) ;
 if (type_ascert == 8 || type_ascert == 9 || type_ascert == 10 ) {
     printf("%6d  %6d   <<the number of discordant/total_concordant
pairs\n",n_DS,n_CS) ;
     printf("%6d  %6d   <<the number of LowConcordant/HighConcordant
pairs\n\n",n_CSL,n_CSH) ;
 }


 ////////////////////////////////////////////////////////////
 //REPORT THE PARAMETERS USED FOR SIMULATION IN THE OUTPUT FILE
 ////////////////////////////////////////////////////////////
fprintf( sib_outfile, "%d\n",-999);
                        /*signal separates the data and the reports*/

fprintf(sib_outfile, "\n%6.3f     <<QTL gene frequency\n",p) ;
```

```
fprintf(sib_outfile, "%6.3f     <<Recombination fraction\n",theta);
fprintf(sib_outfile,"%6d     <<number of marker allele\n ",
         markerCardinality);
fprintf(sib_outfile,"%6.3f %6.3f %6.3f   <<population means for DD/Dd/dd\n",
      mu_DD,mu_Dd,mu_dd);
fprintf(sib_outfile,"%6.3f %6.3f %6.3f    <<population standard deviations
for DD/Dd/dd\n", sd_DD,sd_Dd,sd_dd);
fprintf(sib_outfile, "%6.3f     << shared environmental
correlation\n",environmental_cor);
fprintf(sib_outfile,"%6.3f   %6.3f   << noise parameters : mean, standard
deviation\n",noise_mu, noise_sd);
fprintf(sib_outfile,"%6d     <<ascertainment scheme\n ", type_ascert);
fprintf(sib_outfile, "%6d  %6d   <<yth and xth for selection\n",y,x);
fprintf(sib_outfile, "%6.7f  %6.7f    <<low_tail &
upper_tail\n",low_tail,high_tail);

if (type_ascert == 8 || type_ascert == 9 || type_ascert == 10 ){
     fprintf(sib_outfile, "\nAdditional parameters for the combined
discordant and concordant samples\n");
     fprintf(sib_outfile, "%6d  %6d   <<y2th and x2th for concordance
selection\n",y2,x2);
     fprintf(sib_outfile, "%6.7f  %6.7f    <<low_tail &
upper_tail\n",low_tail2,high_tail2);
     fprintf(sib_outfile, "%6d  %6d   <<the number of
discordant/total_concordant pairs\n",n_DS,n_CS) ;
     fprintf(sib_outfile, "%6d  %6d   <<the number of
LowConcordant/HighConcordant pairs\n\n",n_CSL,n_CSH) ;

   }//match if ascertainment 8-10

fprintf(sib_outfile, "\n\n%8d   <<the number of replicates\n",n) ;
fprintf(sib_outfile,"%8d   <<total # of families (pairs) screened
\n",Number_Fam_Screened);
fprintf(sib_outfile,"%8d   <<total # of families (pairs) ascertained after
examining\n",n);
fprintf(sib_outfile,"%8d   <<total # of people
phenotyped\n",Number_Sibs_Phenotyped);
fprintf(sib_outfile,"%8d   <<total # of people genotyped\n\n",2*n);
if (c_NCP=='y' || c_NCP == 'Y') {
 fprintf(sib_outfile,"%6.3f   << average_NCP_per_sibpair\n\n",average_NCP);
 fprintf(sib_outfile,"%6.3f %6.3f %6.3f %6.3f  <<input population parameters
mean/std/corr/H^2\n",pop_mean,pop_std,pop_sibcorr,overall_heritability);
}
fclose (sib_outfile);


fprintf( noise_outfile, "%d\n",-999);
                        /*signal separates the data and the reports*/

fprintf(noise_outfile, "\n%6.3f     <<QTL gene frequency\n",p) ;
fprintf(noise_outfile, "%6.3f     <<Recombination fraction\n",theta);
fprintf(noise_outfile,"%6d     <<number of marker allele\n
",markerCardinality);
fprintf(noise_outfile,"%6.3f %6.3f %6.3f   <<population means for
DD/Dd/dd\n",mu_DD,mu_Dd,mu_dd);
fprintf(noise_outfile,"%6.3f %6.3f %6.3f    <<population standard deviations
for DD/Dd/dd\n", sd_DD,sd_Dd,sd_dd);
```

```
fprintf(noise_outfile, "%6.3f     << shared environmental
correlation\n",environmental_cor);
fprintf(noise_outfile,"%6.3f   %6.3f   << noise parameters : mean, standard
deviation\n", noise_mu, noise_sd);
fprintf(noise_outfile,"%6d     <<ascertainment scheme\n ", type_ascert);
fprintf(noise_outfile, "%6d  %6d   <<yth and xth for selection\n",y,x);
fprintf(noise_outfile, "%6.7f  %6.7f    <<low_tail &
upper_tail\n",low_tail,high_tail);

 if (type_ascert == 8 || type_ascert == 9 || type_ascert == 10 ) {
     fprintf(noise_outfile, "\nAdditional parameters for the combined
discordant and concordant samples\n");
     fprintf(noise_outfile, "%6d  %6d   <<y2th and x2th for concordance
selection\n",y2,x2);
     fprintf(noise_outfile, "%6.7f  %6.7f    <<low_tail &
upper_tail\n",low_tail2,high_tail2);
     fprintf(noise_outfile, "%6d  %6d <<the number of
discordant/total_concordant pairs\n",n_DS,n_CS) ;
     fprintf(noise_outfile, "%6d  %6d   <<the number of
LowConcordant/HighConcordant pairs\n\n",n_CSL,n_CSH) ;
}

fprintf(noise_outfile, "\n\n%8d <<the number of replicates\n",n) ;
fprintf(noise_outfile,"%8d <<total # of families (pairs)
screened\n",Number_Fam_Screened);
fprintf(noise_outfile,"%8d <<total # of families (pairs) ascertained after
examining \n",n);
fprintf(noise_outfile,"%8d <<total # of people
phenotyped\n",Number_Sibs_Phenotyped);
fprintf(noise_outfile,"%8d <<total # of people genotyped\n\n",2*n);
if (c_NCP=='y' || c_NCP == 'Y') {
  fprintf(noise_outfile,"%6.3f   <<average_NCP_per_sibpair\n",average_NCP);
  fprintf(noise_outfile,"%6.3f %6.3f %6.3f %6.3f  <<input population
parameters
mean/std/corr/overall_heritability\n",pop_mean,pop_std,pop_sibcorr,overall_he
ritability);
}
fclose (noise_outfile);

exit(0);
} //match the main()




//////////////////////////
///subroutine library
//////////////////////////


void ChildGen(long *MatChrom1, long *MatChrom2, long * PatChrom1, long *
PatChrom2,long *Child_Marker, long *Child_DSL, double theta)
{
  /* First, determine chromosome 1 for the child */

  if ( runif() < 0.5 ){
```

```
    if ( runif() >= theta ) {
       Child_Marker[0] = MatChrom1[0]; Child_DSL[0] = MatChrom1[1];
    }
    else {Child_Marker[0] = MatChrom1[0]; Child_DSL[0] = MatChrom2[1];}
  }

  else{
    if ( runif() >= theta ) {
       Child_Marker[0] = MatChrom2[0]; Child_DSL[0] = MatChrom2[1];
    }
    else {Child_Marker[0] = MatChrom2[0]; Child_DSL[0] = MatChrom1[1];}
  }

  if ( runif() < 0.5 ){
    if ( runif() >= theta ) {
       Child_Marker[1] = PatChrom1[0]; Child_DSL[1] = PatChrom1[1];
    }
    else {Child_Marker[1] = PatChrom1[0]; Child_DSL[1] = PatChrom2[1];}

  }

  else{
    if ( runif() >= theta ) {
       Child_Marker[1] = PatChrom2[0]; Child_DSL[1] = PatChrom2[1];
    }
    else {Child_Marker[1] = PatChrom2[0]; Child_DSL[1] = PatChrom1[1];}
  }
}




void quicksort(Item a[], int l, int r)
  { int i;
    int partition(Item a[], int l, int r);
    if (r <= l) return;
    i = partition(a, l, r);
    quicksort(a, l, i-1);
    quicksort(a, i+1, r);
  }



int partition(Item a[], int l, int r)
  { int i = l-1, j = r; Item v = a[r];
    for (;;)
      {
        while (less(a[++i], v)) ;
        while (less(v, a[--j])) if (j == l) break;
        if (i >= j) break;
        exch(a[i], a[j]);
      }

    exch(a[i], a[r]);
    return i;
  }
```

```
Item trait_null( )
   {

    Item trait, noise, total_trait;

       float x= runif();
       if ( x < (p*p))
                 trait = rnorm()*sd_DD + mu_DD;
       else if( x > (1-(1-p)*(1-p)) )
                 trait = rnorm()*sd_dd + mu_dd;

       else
                 trait= rnorm()*sd_Dd + mu_Dd;

   noise = rnorm()*noise_sd + noise_mu;
   total_trait = trait+noise;

   return total_trait;
  }



int select(Item a[],int choice,double trait_sib1,double trait_sib2, float
low, float high, float low2, float high2, int *p_DS, int *p_CSL, int *p_CSH,
int *p_status)

{
    int T = 0;     /*initial value*/
    *p_status = 2; /*initial value*/

//printf("in select () initial *p_status %5d\n",*p_status);

/* Here we select sibs according to user choice and empirical cut-off
values*/
     if ( choice == 0 ) {
                 T=1;

     } else if ( choice == 1 ){ /*single.proband-both.sibs*/
             if((trait_sib1 > high) || (trait_sib2 > high) )
                  T=1;

     } else if ( choice == 13 ){ /*single.proband-one.sib*/
             if((trait_sib1 > high)){
                  T=1;
             }else{
             *p_status = 1; /*T=0*/
             }

     } else if (choice == 2) {
             if((trait_sib1 < low) || (trait_sib2 < low) )
                  T=1;

     } else if (choice == 14) {
             if((trait_sib1 < low)) {
```

```
            T=1;
          }else{
          *p_status = 1; /*T=0*/
          }

} else if (choice == 3) {
          if((trait_sib1 < low) || (trait_sib1 > high) ||
             (trait_sib2 < low) || (trait_sib2 > high) )
               T=1;

} else if (choice == 15) {
          if((trait_sib1 < low) || (trait_sib1 > high)){
               T=1;
          }else{
          *p_status = 1; /*T=0*/
          }

} else if (choice == 4) {
          if( (trait_sib1 > low) && (trait_sib1 < high) )
             *p_status = 1;    /* i.e.T=0, phenotyping_status=1 */
          if( ((trait_sib1 < low) && (trait_sib2 > high)) ||
              ((trait_sib2 < low) && (trait_sib1 > high)) )
               T=1;

} else if (choice == 5 || choice == 11) {
         if (trait_sib1 < high)
                   *p_status = 1;    /* i.e.T=0, phenotyping_status=1 */
        if( (trait_sib1 > high) && (trait_sib2 > high) )
               T=1;

} else if (choice == 6 || choice == 12) {
          if (trait_sib1 > low)
               *p_status = 1;    /* i.e.T=0, phenotyping_status=1 */
          if ((trait_sib1 < low) && (trait_sib2 <  low))
               T=1;

} else if (choice == 7) {
          if( (trait_sib1 > low) && (trait_sib1 < high) )
                *p_status = 1;    /* i.e.T=0, phenotyping_status=1 */
          if( ((trait_sib1 < low) && (trait_sib2 <  low)) ||
              ((trait_sib1 > high) && (trait_sib2 > high)) )
                 T=1;

} else if (choice == 8) {
           if( (trait_sib1 > low) && (trait_sib1 < high) )
                 *p_status = 1;    /* i.e.T=0, phenotyping_status=1 */

           if( ((trait_sib1 < low) && (trait_sib2 > high)) ||
            ((trait_sib2 < low) && (trait_sib1 > high)) )
            {   T=1; (*p_DS)++; }
           else if ( (trait_sib1 > high2) && (trait_sib2 > high2) )
          {   T=1; (*p_CSH)++; }

} else if (choice == 9) {
           if( (trait_sib1 > low) && (trait_sib1 < high) )
               *p_status = 1;    /* i.e.T=0, phenotyping_status=1 */
```

```
              if( ((trait_sib1 < low) && (trait_sib2 > high)) ||
              ((trait_sib2 < low) && (trait_sib1 > high))  )
            {   T=1; (*p_DS)++; }
              else if((trait_sib1 < low2) && (trait_sib2 < low2) )
            {   T=1; (*p_CSL)++; }

     } else if (choice == 10) {
            if( (trait_sib1 > low) && (trait_sib1 < high) )
                *p_status = 1;    /* i.e.T=0, phenotyping_status=1 */


            if( ((trait_sib1 < low) && (trait_sib2 > high)) ||
              ((trait_sib2 < low) && (trait_sib1 > high))  )
              {   T=1; (*p_DS)++; }
            else if ( ((trait_sib1 < low2) && (trait_sib2 < low2))  ||
                  ((trait_sib1 > high2) && (trait_sib2 > high2))  )
            {   T=1;
            if ((trait_sib1 < low2) && (trait_sib2 < low2)) (*p_CSL)++;
                if ((trait_sib1 > high2) && (trait_sib2 > high2))
(*p_CSH)++;

                }// match else if || condition

     }

/*checking codes*/
//printf ("inside: total_sib1 total_sib2 %6.2f
%6.2f\n",trait_sib1,trait_sib2);
//printf ("*p_status %2d\t T_status %2d\n\n", *p_status, T);


   return T;
 }

/* note assuming cut_off_CS more or equally extreme as cutoff_DS*/
```

APPENDIX D

STATISTICAL C++ PROGRAM

```
/*************************************************************************/
/*                            CalSib8.c                                  */
/*                                                                       */
/*                                                                       */
/*A) C++ program CalSib8.c implements various (36) QTL mapping statistics */
/*   for sibling pairs. It can be used to analyze a single dataset or it  */
/*   can be used to evaluate the power of each statistic empirically over */
/*   any number of simulated datasets. Sensitivity study of the effect of */
/*   parameter misspecification can be easily conducted by varying the    */
/*   values of the input parameters.                                     */
/*                                                                       */
/*                                                                       */
/*B) Compiling                                                           */
/*    Step 1) Include "CalSib8.c" AND the three required header files     */
/*            "IBD.c","random.h","simplex.h" in the same working directory*/
/*    Step 2) Use C++ compiler to complie the program by typing:         */
/*            "g++ CalSib8.c -o CalSib8.out"                             */
/*            Note that the name of the executable can be freely changed. */
/*                                                                       */
/*                                                                       */
/*C) A brief description of the QTL mapping statistics implemented:      */
/*                                                                       */
/*    Variance components (e.g. Amos 1994) - Note that it is implemented  */
/*           (see Vc() subroutine) but the function call is commented out,*/
/*           i.e. no results to be output for VC method.                 */
/*    ORIGINAL.HE: regress squared trait difference on IBD-sharing (pi)   */
/*                 and estimate beta_diff (Haseman and Elston 1972)       */
/*    TRAIT.SUM: regress mean-corrected squared trait sum on pi,          */
/*                and estimate beta_sum                                   */
/*    TRAIT.PRODUCT: regress mean-corrected trait product on pi           */
/*                 (Elston et al 2000)                                   */
/*    FORREST:  Weighted average of beat_diff and beta_sum, with weights  */
/*              based on empirical variances of beta_diff and beta_sum    */
/*              (estimation using iterative least square) (Forrest 2001)  */
/*    V&H: Weighted average of beat_diff and beta_sum, with weights       */
/*              based on empirical variances of beta_diff and beta_sum    */
/*              (Visscher and Hopper 2001)                                */
```

```
/*    XU: Weighted average of beat_diff and beta_sum, with weights        */
/*              based on empirical variances and covariance of beta_diff  */
/*              and beta_sum (Xu et al 2000)                              */
/*    S&P1 (HE.COM-correlation) - regress a weighted sum of squared       */
/*          trait difference and sum on pi, with trait values standardized*/
/*          first and weights based on population value of sibling        */
/*          correlation.  (Sham and Purcell 2001)                         */
/*    S&P2 (HE-COM-combination) - regress a function of trait values and  */
/*          trait parameters (Ai as in score3) on pi-1/2, with intercept  */
/*          fixed at zero. Robust variant of S&P1  (Sham and Purcell 2001)*/
/*    SCORE1(Tang and Siegmund 2001): Asymptotic score statistic(standard)*/
/*    SCORE2(Tang and Siegmund 2001): Score statistic with partially      */
/*          empirical variance. (score_robust)                            */
/*    SCORE3 (T.Cuenco et al 2003): with fully empirical variance.        */
/*    SCORE4 (T.Cuenco et al 2003): Score statistic with empirical mean   */
/*           and variance. It is a correlation-based statistic.           */
/*    SCORE5: SCORE3 with pi-bar in the denominator instead of 1/2.       */
/*    SCORE6 (RDP statistic): SCORE3 replacing Ai with squared trait      */
/*           difference.  (Szatkiewicz and Feingold 2004)                 */
/*                                                                        */
/*    IBD1: Risch and Zhang (1995) mean IBD sharing statistics normalized */
/*          by its theoretical variance. One-sided z test.                */
/*          Test at the positive side for concordant pairs; test at the   */
/*          negative side for discordant pairs.                           */
/*    IBD2: standardize pi-bar with empirical variance.                   */
/*    IBD3: IBD2 with pi-bar in the denominator instead of 1/2.           */
/*    IBD1_EDAC(Gu et al. 1996): IBD-sharing statistic for EDAC pairs     */
/*          standardized by theoretical variance.                         */
/*    IBD2_EDAC(Gu-empirical.variance) - IBD1_EDAC with empirical variance*/
/*                                                                        */
/*    Composite statistic for discordant sib pairs (DS) is a weighted sum */
/* of an IBD-sharing statistic (IBD1, IBD2, IBD3) and the original        */
/* Haseman-Elston statistic (ORIGINAL.HE) (note both take negative value):*/
/* COMPOSITE1, COMPOSITE2, COMPOSITE3 (Szakiewicz et al. 2003)            */
/* For EDSP (10%): extreme weights (0.259,0.966)                          */
/* For MDSP (35%): equal weights. (see Forrest and Feingold 2000)         */
/*                                                                        */
/*    Composite statistic for concordant pairs is formed by averaging     */
/* positive signed IBD2/IBD3 with S&P1 statistic (only equal weights      */
/* implemented: COMPOSITE_CS2, COMPOSITE_CS3                              */
/*                                                                        */
/*    Composite statistics for the combined discordant and concordant sib */
/* pairs - two approaches six versions. The first approach compute IBD2   */
/* and regression component statistics for the combind (entire) EDAC      */
/* sample. The second approach first seperates the EDAC sample into two   */
/* groups: the discorant sib pairs (DS) and the concordant sib (CS) pairs,*/
/* then computes IBD-sharing and regression component statistics seperate-*/
/* ly for DS and for CS, and then combine the four (or three) components  */
/* with appropriate weights. See Szatkiewicz and Feingold 2004 for details*/
/* COMPOSITE_DAC11:  IBD2+S&P1                                            */
/* COMPOSITE_DAC12: -IBD2+S&P1                                            */
/*   Note: equal weights are implemented for the above two composite      */
/* COMPOSITE_DAC2:  weighted sum of (IBD2_DS), (ORIGINAL.HE_DS),(IBD_CS)  */
/*                                  (S&P1_CS)                             */
/* COMPOSITE_DAC3:  weighted sum of (IBD2_DS), (ORIGINAL.HE_DS),(IBD_CS)  */
/* COMPOSITE_DAC6:  weighted sum of SCORE6(or RDP)_DS and (IBD_CS)        */
/*   Note: four sets of weights are implemented for the above four        */
```

```
/* composite statistics for the four ascertainment schemes:               */
/* EDAC-3corners (12%, 4%), EDAC-4corners (12%, 4%)                       */
/* MDAC-3corners (24%, 8%), MDAC-4corners(24%, 8%)                       */
/*                                                                        */
/* Maximized composite statistics (weights maximized on particular dataset*/
/* composite_DS4: two components    (on th line of COMPOSITE_DS2)         */
/* COMPOSITE_DAC4: three components (on th line of COMPOSITE_DAC3)        */
/* COMPOSITE_DAC5: four components  (on th line of COMPOSITE_DAC2)        */
/*   Note: each of above statistics is mixture of chi-squared variables.  */
/*                                                                        */
/*                                                                        */
/*                                                                        */
/*D) Some checking are done to insure that input files have               */
/*values that fall within the correct range.                              */
/*   The number of families to be looked at must be greater than zero.    */
/*   The program does not check whether or not IBD values for a family    */
/*sum to 1.000.                                                           */
/*                                                                        */
/*                                                                        */
/*                                                                        */
/*E) Program Inputs:                                                      */
/*   "CalSib8.c" requires the following  three input files:               */
/*    "sib_simualte", "CalSib8_input_file", "population_file"             */
/*    all to be included in teh same working directory.                   */
/*                                                                        */
/*  (1) "sib_simualte" stores the trait values and IBD-sharing information*/
/*     of each sibling pair. The output file "sib_simualte" generated by  */
/*     our simulation program ("Newsimsib5.c" and "Newsimsib_nonnormal.c" */
/*     is ready to use. If analyzing any out-sources dataset which is not */
/*     generated by any of our simulaiton program, the data matix should  */
/*     be formatted and named exactly the same as "sib_simulate".         */
/*                                                                        */
/*  (2) CalSib8_input_file" contains the following three lines:           */
/*         100       << line 1: number of pairs in each dataset (num_fam) */
/*         1000      << line 2: number fo datasets (N_sample)             */
/*         112       << line 3: choice code for how to input trait        */
/*                      parameters that are required by some statistics   */
/*     Note for the choice code in line 3                                 */
/*         112 - choise "p", i.e. input from input file "population_file";*/
/*         117 - choise "u", i.e. input by users from screen at unixs $  ;*/
/*         115 - choise "s", i.e. calculate sample estimates of parameters*/
/*                                                                        */
/*   (3) If the choice is 112, i.e. input from file "population_file",    */
/*       you must inlcude this file as input file in the same directory.  */
/*       "population_file" contains the following three lines:            */
/*         -0.8      << overall trait mean                              */
/*          0.9      << overall trait variance                          */
/*          0.3      << overall sibling correlation                     */
/*                                                                        */
/*    Note that a minor output file called "user_COMPOSITE" is also       */
/* produced. This file is useful ONLY if you want to explore weights for  */
/* various composite statistics for EDAC samples ascertained using        */
/* arbitrary selection thresholds (see below file pointer fp9).           */
/*                                                                        */
/*                                                                        */
/*F) Program Outputs:                                                     */
/*    "CalSib8.c" produces three output files:                            */
```

```
/*    "user_parameters","statistics" and "powers".                          */
/*  (1) The file "statitics" is a data matrix of 36 (total number of       */
/*      statistics by N_samples (total number of dataset), where each      */
/*      column is an array of one of the 36 QTL mapping statistics. This   */
/*      file can be imported to any statistical package for further        */
/*      examination. These statistics are:                                 */
/*      - For any of the regression methods: t-statistic of the slope      */
/*      - For any of the score statistics and variants: z-score            */
/*      - For any of the IBD-sharing statistics: z- score                  */
/*      - For any of the composite statistics: z-score                     */
/*      - For any of the maximized composite statistics: chi-squared       */
/*      - For variance components: -2LL                                    */
/*                                                                         */
/*   (2) The file "powers" consists of four columns                        */
/*       column 1: the lsit of the 36 statistics                           */
/*       column 2: empirical power of each statistic evaluated over all    */
/*                 (N_samples) of the datasets                             */
/*       column 3: the mean of each statistic                              */
/*       column 4: the standard deviation of each statistic                */
/*                                                                         */
/*   (3) "user_parameters" stores the value of num_fam, N-samples, and     */
/*      the choise code for input,i.e. the contents of "CalSib8_input_file*/
/*      it serves as a reminder/checking.                                  */
/*                                                                         */
/*                                                                         */
/*G) Primary variables:                                                    */
/*   num_fam , N-samples;                                                  */
/*   trait1, trait2, IBDprob0, IBDprob1, IBDprob2;                         */
/*   IBD, est_IBD;                                                         */
/*   sqr_trait_diff, sqr_trait_sum, product;                               */
/*   trait_mean. trait_var, trait_cov, corr, std_dev_trait;                */
/*   standardizedtrait1, standardizedtrait2;                               */
/*   outcome_Sham,outcome_robustSham(Ai),score_c(Ai/4 func. in papers);    */
/*   original_HE, new_HE,sum_HE,Forrest_HE, Xu_HE, Sham_HE, robust_Sham,   */
/*score_standard, score_robust, score_3,score_4,score_5, score_6, LL_store*/
/*meanIBD.meanIBD2. meanIBD3, IBD1_EDAC,IBD2_EDAC,                         */
/*    composite1, composite2, composite3 (for DS), composite_DS4;          */
/*composite_CS2, composite_CS3 (for CS);  COMPOSITE_DAC11,COMPOSITE_DAC12.*/
/* COMPOSITE_DAC2. COMPOSITE_DAC3. COMPOSITE_DAC6.                         */
/* COMPOSITE_DAC4.COMPOSITE_DAC5                                           */
/*    The components computed for EDAC sample:                             */
/* IBD2_DS, IBD2_CS, HE_DS (original.HE for DS), HE_CS (S&P1 for CS),      */
/* HE_CS2 (original.HE for CS), score6_DS (RDP for DS)                     */
/*    fp1="population_file",fp2="sib_simualte",                            */
/*    fp4="statistics", fp5="powers".                                      */
/*    fp9="user_COMPOSITE", it can be used for output of the components    */
/* computed for EDAC samples based on arbitrary thresholds. If used, this */
/* output file can then be imported to R in order to explore appropriate   */
/* weights associated with that selection scheme.                          */
/*                                                                         */
/*                                                                         */
/*                                                                         */
/*H) Subroutines:                                                          */
/*   1. regression():outcome variable and independent variable are passed;*/
/*standard linear regression is performed, which gives the estimated value*/
/*of the slope & intercept of the regression line and their standard error*/
/*     regression() returns the standardized slope (T-statistic) to main()*/
```

```
/* Depending upon different outcome variable, it calculates different     */
/* T-statistics, such as original H-E, sum H-E, new H-E and Sham H-E.     */
/*                                                                        */
/*   2. Forrest() returns standardized Forrest weighted H-E slope         */
/*   3. Xu()returns standardized Xu weighted H-E slope                    */
/*Xu() also computes standardized Visscher&Hopper weighted H-E slope      */
/*   4. robustShamreg() performs regression with no intercept; and it     */
/*returns  robust_Sham H-E slope(T-statistic)                            */
/*   5. Tang() calculates score1-6 and return the values to main().       */
/*   6. Vc() performs variance component method and returns -2LL          */
/*   7. power() calculates the mean, standard deviation and the empirical */
/*power  for each statistic. Critical values of each test is hard coded/  */
/*predetermined. It generates the output file "powers".                  */
/*                                                                        */
/*                                                                        */
/*                                                                        */
/*I) The code for ORIGINAL.HE,TRAIT.SUM,TRAIT.PRODUCT,FORREST,IBD1,       */
/* COMPOSITE1 and variance components is taken from programs by           */
/*Dr. William F. Forrest III (Forrest and Feingold 2000)                  */
/*forrest@forrest.hgen.pitt.edu dated: Mon, 18 Oct 1999.                  */
/*                                                                        */
/*                                                                        */
/*                                                                        */
/*J) The associated header files and simulation program                   */
/*   Here this is the main C++ script to calculate  all the statistics and*/
/*to find powers for all the tests. The header file "simplex.h" has a     */
/*minimization routine taken from  "Numerical Recipes in C".              */
/*   The file "IBD.c" and The file "random.h" are used by simulation      */
/*program "simsib.c"                                                      */
/*   This program requires C++ compiler. Type g++ to compile.             */
/*************************************************************************/




//opening up libraries and header files needed for this program
#include<stdlib.h>
#include<stdio.h>
#include<math.h>
#include<limits.h>
#include<time.h>
#include<iostream.h>
#include "random.h"
#include "IBD.c"
#include "simplex.h"
#include<assert.h>
#include<iomanip.h>


// GLOBAL variables are defined here for convenience in
// the variance components optimization later on.

long N_samples;   /*# of studies; unknown until run_time*/
long num_fam;     /*per study; unknown until run_time*/
//heap memory & dynamic allocation will be used in main()
double * trait1;  /* Trait value for sibling 1 */
```

```
double * trait2;  /* Trait value for sibling 2 */
double * est_IBD; /* Estimated IBD for a pair */


double mu, sig_e, lambda, sig_g;
double loglike( double , double , double , double ,
               const double * , const double * , const double * ,
                long );
double LL_mu(double);
double LL_sig_e(double);
double LL_lambda(double);
double LL_sig_g(double);
double LL_res(int,double*);
double LL_full(int, double *);
double K3(int, double *);
double K4(int, double *);
double K4b(int, double *);


main()   {


  ////////////////////////////////////
  //Variable declarations
  ////////////////////////////////////
int c=0 ; /*character input*/
long i=0; /*counter for looping*/
long j=0; /*counter for N_samples */

double trait_mean, trait_var, corr,trait_cov;
double top1;
double top2;
double std_dev_trait;

long total_n_DS=0, total_n_CS=0;


//////////////////////////////////////////
//initialize file pointers for input files//
//////////////////////////////////////////
FILE * inputN; /* file pointer for "CalSib8_input_file"*/
FILE *fp1 ;/*file pointer for "population_file" */
FILE *fp2 ;/*file pointer for "sib_simualte"*/

//////////////////////////////////////////
//initialize file pointers for output files//
//////////////////////////////////////////
FILE * user_parameter;
user_parameter = fopen("user_parameters", "w");
FILE *fp4; /*file pointer for outputted statistics*/
fp4=fopen("statistics", "w");
/* print heading for the file - name of each statistic*/
fprintf(fp4,"original_HE product_HE sum_HE Forrest_HE Xu_HE Viss_HE S&P1 S&P2
score1 score2 score3 score4 score5 RDP IBD1 IBD2 IBD3 IBD1_EDAC IBD2_EDAC
COMPOSITE_DS1 COMPOSITE_DS2 COMPOSITE_DS3 COMPOSITE_DS4 COMPOSITE_CS2
COMPOSITE_CS3 COMPOSITE_DAC11 COMPOSITE_DAC12 COMPOSITE_DAC2 COMPOSITE_DAC3
COMPOSITE_DAC6 COMPOSITE_DAC4 COMPOSITE_DAC5 LL_store\n");
```

```
/* the file below is only useful if you want to explore weights*/
/* for composite_DAC, note that header=True*/
FILE *fp9; /*file pointer for components of COMPOSITE_DAC*/
fp9=fopen("user_COMPOSITE", "w");
fprintf(fp9,"original.HE_DS IBD2_DS HE.COM1_CS IBD2_CS original.HE_CS
RDP_DS\n");


////////////////////////////////////////////////
//input  for num_fam & N-samples and choice code
////////////////////////////////////////////////
if ((inputN = fopen("CalSib8_input_file", "r")) == NULL) {
   fprintf(stderr,"Can not open input file 'CalSib8_input_file'\n\n");
   exit(13);
} else {
   fscanf(inputN,"%d", &num_fam);
   if (num_fam < 1){
                printf("This file has zero families.\n");
                exit(1);
                }
   fscanf(inputN,"%d", &N_samples);
   if (N_samples<1){
             printf("You can not have zero studies.\n");
                exit(2);
            }
   fprintf(user_parameter, "num_fam %d N_samples %d\n",  num_fam, N_samples);

   fscanf(inputN, "%d",&c);
   printf("%c\t%d\n",c,c);
   fprintf(user_parameter, "choice of population_file, user_input or sample
estimates: %c\n",c);
   fclose(inputN);

}



////////////////////////////////////////////////////////
//more variable declaration after input of num_fam & N-samples
////////////////////////////////////////////////////////

trait1 = (double *) calloc(num_fam,sizeof(double));
trait2 = (double *) calloc(num_fam,sizeof(double));
est_IBD = (double *) calloc(num_fam,sizeof(double));

if (trait1==NULL ||trait2==NULL||est_IBD==NULL){
      printf("\nUnable to allocate %d element in one study\n",num_fam);
        exit(6);
}



double * IBDprob0 = new double [num_fam];
double * IBDprob1 = new double [num_fam];
double * IBDprob2 = new double [num_fam];
double * IBD = new double [num_fam];
double * meanIBD1 = new double[N_samples];
```

```
double * meanIBD2 = new double [N_samples];
double * meanIBD3 = new double [N_samples];
double *IBD1_EDAC = new double [N_samples];
double *IBD2_EDAC = new double [N_samples];

double * composite1 = new double [N_samples];    /*COMPOSITE_DS1*/
double * composite2 = new double [N_samples];    /*COMPOSITE_DS2*/
double * composite3 = new double [N_samples];    /*COMPOSITE_DS3*/
double * composite_CS2 = new double [N_samples];
double * composite_CS3 = new double [N_samples];

double * sqr_trait_sum = new double [num_fam];
double * sqr_trait_diff = new double [num_fam];
double * trait_diff = new double [num_fam];
double * product = new double [num_fam];
double * meancorrectedtrait1 = new double [num_fam];
double * meancorrectedtrait2 = new double [num_fam];
double * standardizedtrait1 = new double [num_fam];
double * standardizedtrait2 = new double [num_fam];
double * outcome_Sham = new double [num_fam];
double * outcome_robustSham = new double [num_fam];

double * original_HE = new double [N_samples];   /*ORIGINAL.HE*/
double * new_HE = new double [N_samples];        /*TRAIT.PRODUCT*/
double * sum_HE = new double [N_samples];        /*TRAIT.SUM*/
double * Forrest_HE = new double [N_samples];    /*FOREEST*/
double * Xu_HE  = new double [N_samples];        /*XU*/
double * Viss_HE  = new double [N_samples];      /*V&H*/
double * Sham_HE = new double [N_samples];       /*S&P1*/
double * robust_Sham = new double [N_samples];   /*S&P2*/
double *score_standard = new double[N_samples];  /*SCORE1*/
double *score_robust = new double[N_samples];    /*SCORE2*/
double *score_3 = new double[N_samples];
double *score_4 = new double[N_samples];
double *score_5 = new double [N_samples];
 double *score_6 = new double [N_samples];     /* RDP statistic*/
double *score_c = new double[num_fam];       /*Ai function in papers*/
double *score_D = new double[num_fam];       /*outcome in original HE*/
double * LL_store = new double [N_samples];

/* component statistics for IBD-sharing and composite statistics for EDAC
samples*/
double *IBD2_DS = new double [N_samples];
double *IBD2_CS = new double [N_samples];
double *HE_DS= new double [N_samples];
double *HE_CS = new double [N_samples];
double *HE_CS2 = new double [N_samples];
double *score6_DS = new double [N_samples];

double *COMPOSITE_DAC11 = new double [N_samples];
double *COMPOSITE_DAC12 = new double [N_samples];
double *COMPOSITE_DAC2 = new double [N_samples];
double *COMPOSITE_DAC3 = new double [N_samples];   /* Three components*/
double *COMPOSITE_DAC6 = new double [N_samples];   /* along with score6*/

double *composite_DS4 = new double [N_samples];/*COMPOSITE_DS4, two
componets*/
```

```
double *COMPOSITE_DAC4 = new double [N_samples];/*M3 -3 components maximized
*/
double *COMPOSITE_DAC5 = new double [N_samples];/*M4 -4  componets maximized
*/


//////////////////////////
//subroutine declaration
//////////////////////////

double regression(double * , long , double * );

double Vc(double *trait1, double *trait2, double *est_IBD,
          double trait_mean, double trait_var, double trait_cov,
          long num_fam);

double Xu(double *sqt_trait_sum, double *trait_diff, long num_fam,
          double *IBD, double *);

double Forrest(double * sum_HE,double *sqr_trait_diff,
               double *sqr_trait_sum,
               long num_fam,   double *est_IBD);

double robustShamreg(double * outcome_robustSham, long num_fam,
                     double * est_IBD);

double Tang(double *,double *, long num_fam, double *est_IBD, double corr,
double* ,double *,double *,double*, double*,double *);

void power (double *LL_store, double *original_HE, double *sum_HE,
          double *new_HE, double *Xu_HE,double *Viss_HE, double *Forrest_HE,
              double *Sham_HE,double *robust_Sham,
              double *score_standard, double *score_robust,
          double *score_3,double *score_4, double *score_5, double *score_6,
              double *meanIBD1,double *meanIBD2, double *meanIBD3,
              double *composite1,double *composite2,double *composite3,
              double *composite_CS2, double *composite_CS3,
              double *COMPOSITE_DAC11, double *COMPOSITE_DAC12,
              double *COMPOSITE_DAC2, double *COMPOSITE_DAC3,
              double *composite_DS4, double *COMPOSITE_DAC4,
              double *COMPOSITE_DAC5,double *COMPOSITE_DAC6,
              double *IBD1_EDAC, double *IBD2_EDAC);




//////////////////////////////////////////////////////////
//if choose to use population parameters, data are read in
//from file 'population_file'
//////////////////////////////////////////////////////////
if (c==112){

      fp1 = fopen("population_file", "r"); /*population parameter file */
        if ((fp1 = fopen("population_file", "r")) == NULL) {
              fprintf( stderr, "Can not open 'population_file'\n\n");
              exit(3);
```

```
        }else{
              fscanf(fp1, "%lf %lf %lf\n", &trait_mean, &trait_var, &corr);
            fprintf(user_parameter, "mean %6.4f variance %6.4f corr %6.4f\n",
trait_mean, trait_var,corr);
       }
  fclose(fp1);
  fclose(user_parameter);
}


////////////////////////////////////////////////////////
//if choose to input parameters at the run time,
//ask users for these parameters
////////////////////////////////////////////////////////
 else if (c==117 ){
   printf("\nInput parameters for model \n");
   printf("\ntrait mean:\n");
   scanf("%lf", &trait_mean);

   printf("\ntrait variance:\n");
   scanf("%lf", &trait_var);

   printf("\ntrait correlation:\n"); //correlation between sibs
   scanf("%lf", &corr);

   fprintf(user_parameter, "%lf %lf %lf\n", trait_mean, trait_var, corr);
   printf("The following paraemters has been read: %lf %lf %lf\n\n",
trait_mean, trait_var, corr);
   fclose(user_parameter);
 }




////////////////////////////////////////////////////
//if choose to use sample statistics
//need to calculate sample parameters in next section
////////////////////////////////////////////////////
else if (c==115){
            fclose(user_parameter);
}



////////////////////////////////////////////////////
//Data file sib_simulate is read in here
//store in trait1[],trait2[],IBD0[],IBD1[],IBD2[]
////////////////////////////////////////////////////

if ( (fp2 = fopen("sib_simulate", "r")) == NULL){
          fprintf(stderr,"Can not open 'sib_simualte'\n\n");
          exit(4);
 }


////////////////////////////////////////////////
//starting the major loop on studies with index j
```

```
//////////////////////////////////////////////
else{
      for (j=0; j <N_samples; j++){     /*loop over all studies*/

         /*******************************************************/
           /*read and store data for 1 study:                     */
           /*2 trait values and 3 IBD values for each family;     */
           /*num_fam: the sample size for each study with index i  */
           /*******************************************************/
            for (i = 0; i< num_fam; i++) {

            /*make sure that it is not the end of data file -999*/
            fscanf(fp2, "%lf\n", &trait1[i]);
            /*stop this num_fam loop if there are no more data, */
                /* other wise proceed reading file.*/
                      if (trait1[i] == -999){
                          printf("\nThis is the end of data file.\n\n");
                          break;
                       }

                fscanf(fp2, "%lf %lf %lf %lf\n",
&trait2[i],&IBDprob0[i],&IBDprob1[i], &IBDprob2[i]);
             }


      if (trait1[i] == -999){
            printf("This is the end of the data file.\n\n");
            printf("%d families were actually read\n",j);
            break;/*stop the loop if there are no more data*/
       }


//////////////////////////////////////////////////////////////
//calculate the sample trait parameters if user choose that option
//////////////////////////////////////////////////////////////
if (c==115){
      trait_mean = 0.0;
      for (i = 0; i< num_fam; i++) {
             trait_mean += (trait1[i]  + trait2[i])/(num_fam*2);
       }

      trait_var = 0.0;
        top1=0.0;
      for (i = 0; i< num_fam; i++) {
trait_var += (((trait1[i] -trait_mean) * (trait1[i] -trait_mean))+
((trait2[i] -trait_mean) * (trait2[i] -trait_mean)))/(2*num_fam-1);
      top1 += trait1[i]* trait2[i];
      }
      top2 = top1/(num_fam) - (trait_mean * trait_mean);


      /*calculating correlation between std.trait1 & std.trait2 */
      corr = top2/(trait_var);
} /*match (if c==115)*/

trait_cov=0.0;
trait_cov = corr * trait_var;
```

```
////////////////////////////////////////////////////
//calculation of summary statistics used for subroutines
//estimated_IBD (est_IBD[i]) and IBD[i];
//
// Calulation of 3 versions of mean IBD sharing staitsics
////////////////////////////////////////////////////

double meanIBD = 0.0;
for (i = 0; i< num_fam; i++) {
    /*normalized*/
    est_IBD[i] = (IBDprob1[i] / 2 + IBDprob2[i]) - 0.50;
    /*un-normalized, will be passed to Xu()*/
    IBD[i] = (IBDprob1[i]/2) + IBDprob2[i] ;

    meanIBD +=est_IBD[i];
}

meanIBD /= (double)num_fam;
/*meanIBD is temporary - the average normalized IBD for the study*/

      double sum_IBD_dev_half = 0.0;
        double sum_IBD_dev_meanIBD= 0.0;
      for(i=0;i<num_fam;i++){
        sum_IBD_dev_half += est_IBD[i]*est_IBD[i];
        sum_IBD_dev_meanIBD += (est_IBD[i]-meanIBD)* (est_IBD[i]-meanIBD);
      }


/*standard meanIBD statistic is the average zeroed IBD for the study,    */
/*then normalized by the theoretical variance = meanIBD[j] * sqrt(8*num_fam)
*/
/*meanIBD2 normalizes by its empirical variance, meanIBD3-fully empirical*/
meanIBD1[j] = meanIBD* sqrt(8*num_fam);
meanIBD2[j] = meanIBD * num_fam / sqrt(sum_IBD_dev_half);
meanIBD3[j] = meanIBD * num_fam / sqrt(sum_IBD_dev_meanIBD);



////////////////////////////////////////////////////////////
//calculation of regression outcome variables for various HE
////////////////////////////////////////////////////////////
//double sumsq_sqr_trait_diff = 0.0;
for (i = 0; i< num_fam; i++) {

    sqr_trait_sum[i] = (trait1[i]+trait2[i]-
2*trait_mean)*(trait1[i]+trait2[i]-2*trait_mean);
    sqr_trait_diff[i] = (trait1[i]-trait2[i]) * (trait1[i]-trait2[i]);
    product[i] =  (trait1[i] -trait_mean)*(trait2[i] -trait_mean);
}



////////////////////////////////////////////////////////////
//standardize traits using the trait_mean and std_dev_trait
```

```
////////////////////////////////////////////////////////
std_dev_trait = sqrt(trait_var);
for (i = 0 ; i < num_fam; i++) {
    meancorrectedtrait1[i] = trait1[i] - trait_mean;
    meancorrectedtrait2[i] = trait2[i] - trait_mean;
    standardizedtrait1[i] = meancorrectedtrait1[i] / std_dev_trait;
    standardizedtrait2[i]  = meancorrectedtrait2[i] / std_dev_trait;
}



////////////////////////////////////////////////////////
//calculate regression outcome variables for SHAM/robust_SHAM
////////////////////////////////////////////////////////
 for (i = 0 ; i < num_fam; i++){
    /***Sham uses standardized outcome variable*/
    outcome_Sham[i] = (((standardizedtrait1[i] + standardizedtrait2[i]) /
                  (1 + corr) )  *
                   ((standardizedtrait1[i] + standardizedtrait2[i]) /
                  (1 + corr) )) -
      (((standardizedtrait1[i] - standardizedtrait2[i]) / (1 - corr))*
       ((standardizedtrait1[i] - standardizedtrait2[i]) / (1 - corr)  ) );

    /***robust_Sham uses standardized outcome variables*/
    outcome_robustSham[i] = outcome_Sham[i] + ((4*corr)/(1-(corr * corr)));

  }


////////////////////////////////////////
//calculation for Ai related funcion used in score_test
////////////////////////////////////////
 for(i=0;i<num_fam;i++)
   { score_c[i] = outcome_robustSham[i]/4.0;  /* for score 1-5*/
     score_D[i] = (sqr_trait_diff[i])/4.0;    /* for score 6 or RDP*/
   }



////////////////////////////////////////
//calls to regression for variants of HE//
////////////////////////////////////////
original_HE[j] = regression(sqr_trait_diff, num_fam, est_IBD);

new_HE[j] = regression(product, num_fam, est_IBD );

sum_HE[j] = regression(sqr_trait_sum, num_fam, est_IBD);

Forrest_HE[j] =
Forrest(&sum_HE[j],sqr_trait_diff,sqr_trait_sum,num_fam,est_IBD);

Xu_HE[j] = Xu(sqr_trait_sum, sqr_trait_diff, num_fam,IBD,&Viss_HE[j]);

Sham_HE[j] = regression(outcome_Sham, num_fam, est_IBD);

robust_Sham[j] = robustShamreg(outcome_robustSham, num_fam, est_IBD);
```

```
Tang(score_c, score_D, num_fam, est_IBD, corr, &score_standard[j],
&score_robust[j], &score_3[j], &score_4[j], &score_5[j], &score_6[j]);

/* no results will be output for Variance components */
//LL_store[j] = Vc(trait1, trait2, est_IBD,trait_mean, trait_var,trait_cov,
num_fam);



////////////////////////////////////
///COMPOSITE statistics for DS or CS
////////////////////////////////////


/*composite statistic for discordant pairs*/
/* weighted sum of IBD sharing statistic and original.HE*/
/*equal weights*/
// composite1[j] = (original_HE[j]+ meanIBD1[j])/sqrt(2);
// composite2[j] = (original_HE[j]+ meanIBD2[j])/sqrt(2);
// composite3[j] = (original_HE[j]+ meanIBD3[j])/sqrt(2);

 /*extreme weights: (0.259,0.966)*/
composite1[j] = 0.259*original_HE[j] + 0.966*meanIBD1[j];
composite2[j] = 0.259*original_HE[j] + 0.966*meanIBD2[j];
composite3[j] = 0.259*original_HE[j] + 0.966*meanIBD3[j];

/*composite statistic for concordant pairs is formed by weighted average of
*/
/* IBD2 or IBD3 with Sham_HE (S&P1) - the best regression statistic for
concordant */
/* equal weights*/
 composite_CS2[j] = (Sham_HE[j]+ meanIBD2[j])/sqrt(2);
 composite_CS3[j] = (Sham_HE[j]+ meanIBD3[j])/sqrt(2);



////////////////////////////////////////////////
//COMPOSITE FOR COMBINED DISCORDANT & CONCORDANT
////////////////////////////////////////////////

/*APPROACH1: apply on entire sample*/
/* Note: if discordant pairs are dominant in the sample, meanIBD for the*/
/* combined EDAC will be negative & COMPOSITE_DAC12 will work in this case*/
/* equal weights*/
COMPOSITE_DAC11[j] = (Sham_HE[j]+ meanIBD2[j])/sqrt(2);
COMPOSITE_DAC12[j] = (-Sham_HE[j]+ meanIBD2[j])/sqrt(2);

/*********************************************************************/
/* APPROCH2: first seperate the sample into DS group and and CS group,  */
/* then compute component statistics for each group seperately, finally */
/* combine components with appropriate weights.                         */
/* Note: product[i] =  (trait1[i] -trait_mean)*(trait2[i] -trait_mean ) */
/* Given that the sample is a combined discordant and concordant sibling*/
/*pair sample, if product<0, then this pairs are discordant; otherwise, */
/*this pair is concordant                                               */
/* For discordant pairs, the indicator of discordance is set to 1,T[i]=1*/
/* o.w, T[i]=0                                                          */
```

```
/*************************************************************************/

//////////////////////////////////////////////////////
/* calculate seperate IBD components: IBD2_DS and IBD2_CS*/
/* these components are also needed for IBD-sharing      */
/* statistics for EDAC samples                           */
//////////////////////////////////////////////////////
int * T = new int [num_fam];
int n_DS = 0, n_CS=0;

double meanIBD_DS=0.0, meanIBD_CS=0.0;
double varIBD_DS=0.0, varIBD_CS=0.0;

for (i = 0; i< num_fam; i++) {

     if (product[i]<0)  {
                    T[i]=1;
                    n_DS++;
                    meanIBD_DS +=est_IBD[i];
                    varIBD_DS += est_IBD[i]*est_IBD[i];
      } else{
                    T[i]=0;
                    n_CS++;
                    meanIBD_CS +=est_IBD[i];
                    varIBD_CS += est_IBD[i]*est_IBD[i];
     }
}//match for

/*if DS only sample,IBD2_DS=meanIBD2[j]*/
/*if CS only sample,IBD2_DS=meanIBD2[j]*/
/*Note, here meanIBD_CS and varIBD_CS both are sums, not devided by n yet*/
 if ((n_DS==0) || (varIBD_DS==0.0))  {IBD2_DS[j]= 0.0;}
 else if ((n_DS>0) && (varIBD_DS>0.0))
     {IBD2_DS[j] = meanIBD_DS/ sqrt(varIBD_DS);}
 if ((n_CS==0) || (varIBD_CS==0.0))  {IBD2_CS[j]= 0.0;}
 else if ((n_CS>0) && (varIBD_DS>0.0))
     {IBD2_CS[j] = meanIBD_CS/ sqrt(varIBD_CS);}

//////////////////////////////////////////////////////////////////////
/*IBD1_EDAC is Gu et al. 1996 version, with theoretical variance 1/8  */
/*IBD2_EDAC replace 1/8 with pooled estimate of variance of two samples*/
/* assuming equal varaince                                          */
//////////////////////////////////////////////////////////////////////
 double quantity = 0.0;
 if ((n_CS==0)&&(n_DS>0)) {quantity = (-meanIBD_DS/n_DS)*sqrt(n_DS); }
 else if ((n_DS==0)&&(n_CS>0)) {quantity = (meanIBD_CS/n_CS)*sqrt(n_CS);}
 else if ((n_CS>0) && (n_DS>0)) {quantity = (meanIBD_CS/n_CS-
meanIBD_DS/n_DS)*sqrt((n_CS*n_DS)/(n_CS+n_DS));}
 else exit(10);

 IBD1_EDAC[j] = quantity*sqrt(8);
 IBD2_EDAC[j] = quantity/sqrt((varIBD_CS+varIBD_DS)/(n_CS+n_DS));


 ////////////////////////////////////////////////////////////
 /* compute seperate regression components HE_DS, HE_CS, HE_CS2*/
 /* First use separate arrays to hold each type of pairs       */
```

```
//////////////////////////////////////////////////////////
int p=0,q=0;
double * outcome_DS = new double [n_DS];
double * outcome_CS = new double [n_CS];
double * outcome_CS2 = new double [n_CS]; /* for COMPOSITE_DAC3*/
double * est_IBD_DS = new double [n_DS];
double * est_IBD_CS = new double [n_CS];
double numerator_DS = 0.0;
double sumsq_sqr_trait_diff_DS = 0.0;

for (i = 0; i< num_fam; i++) {
   if (T[i]==1)
    {   outcome_DS[p]= sqr_trait_diff[i]; est_IBD_DS[p] = est_IBD[i];
        numerator_DS += outcome_DS[p]*est_IBD_DS[p];
        sumsq_sqr_trait_diff_DS += outcome_DS[p]*outcome_DS[p];
    p++;
    }else if (T[i]==0)
      {   outcome_CS[q]= outcome_Sham[i];
          outcome_CS2[q]= sqr_trait_diff [i]; /* for 3-part-composite*/
          est_IBD_CS[q] =est_IBD[i];
          //sum_sqr_trait_diff_CS +=outcome_CS2[q];
    q++;
    }//match if
}//match for

HE_DS[j] = regression(outcome_DS, n_DS, est_IBD_DS);
HE_CS[j] = regression(outcome_CS, n_CS, est_IBD_CS);
HE_CS2[j] = regression(outcome_CS2, n_CS, est_IBD_CS);
score6_DS[j] = numerator_DS/sqrt(sumsq_sqr_trait_diff_DS*varIBD_DS/n_DS);
/* RDP for discordant pairs*/

//////////////////////////////////////////////////////////
/* combine components with appropriate weights         */
/* Applying four sets of weights for each ascertainment*/
/* see Szatkiewicz adn Feingold 2004 for details       */
//////////////////////////////////////////////////////////

/*EDAC-3corner, 12% for DS and 4% for CS*/
//COMPOSITE_DAC2[j]= -0.2467*HE_DS[j]-
0.7077*IBD2_DS[j]+0.1619*HE_CS[j]+0.6420*IBD2_CS[j];
//COMPOSITE_DAC3[j]= -0.239*HE_DS[j]-0.685*IBD2_DS[j]-
0*HE_CS2[j]+0.688*IBD2_CS[j];
//COMPOSITE_DAC6[j] = -0.726*score6_DS[j] + 0.688*IBD2_CS[j]; /* 43 degree */

/*EDAC-4corner, 12% for DS and 4% for CS*/
//COMPOSITE_DAC2[j]= -0.2427*HE_DS[j]-
0.6965*IBD2_DS[j]+0.2666*HE_CS[j]+0.6204*IBD2_CS[j];
//COMPOSITE_DAC3[j]= -0.254*HE_DS[j]-0.729*IBD2_DS[j]-
0*HE_CS2[j]+0.635*IBD2_CS[j];
//COMPOSITE_DAC6[j] = -0.772*score6_DS[j] + 0.635*IBD2_CS[j]; /* 39 degree */

/*MDAC-3corner, 24% and 8% */
//COMPOSITE_DAC2[j]= -0.3884*HE_DS[j]-
0.6129*IBD2_DS[j]+0.1918*HE_CS[j]+0.6609*IBD2_CS[j];
//COMPOSITE_DAC3[j]= -0.436*HE_DS[j]-0.689*IBD2_DS[j]-
0*HE_CS2[j]+0.579*IBD2_CS[j];
//COMPOSITE_DAC6[j] = -0.815*score6_DS[j] + 0.579*IBD2_CS[j]; /* 35 degree */
```

```
/*MDAC-4corner, 24% and 8% */
COMPOSITE_DAC2[j]= -0.4394*HE_DS[j]-
0.7214*IBD2_DS[j]+0.2114*HE_CS[j]+0.4917*IBD2_CS[j];
COMPOSITE_DAC3[j]= -0.462*HE_DS[j]-0.759*IBD2_DS[j]-
0*HE_CS2[j]+0.4587*IBD2_CS[j];
COMPOSITE_DAC6[j] = -0.889*score6_DS[j] + 0.458*IBD2_CS[j]; /* 27 degree */

 /*Applying equal weights*/
 //COMPOSITE_DAC2[j]= (-HE_DS[j]-IBD2_DS[j]+HE_CS[j]+IBD2_CS[j])/2;
 //COMPOSITE_DAC3[j]= (-HE_DS[j]-IBD2_DS[j]-HE_CS2[j]+IBD2_CS[j])/2;
 //COMPOSITE_DAC6[j] = (-score6_DS[j] + IBD2_CS[j])/sqrt(2);

 /* output to file "user_COMPOSITE", used if want to explore*/
 /* weights for other different ascertainment schemes       */
 /* R program for searching for optimal weights are included*/
 /* in the dissertation by Jin Szatkiewicz, Univ. of Pitt 2004*/
 fprintf(fp9, "%lf %lf %lf %lf %lf %lf\n",HE_DS[j],IBD2_DS[j], HE_CS[j],
IBD2_CS[j],HE_CS2[j],score6_DS[j]);



//////////////////////////////////////////////////
// Maximized composite statistics - one sided test
// Distribtuions are mixture of chi-sq
///////////////////////////////////////////////////

int w1=0, w2=0, w3=0, w4=0 ;/*indicator set to 1 if the component is
positive*/

if (-original_HE[j]>0) { w1=1;}
if (-meanIBD2[j]>0) {w2=1;}
 composite_DS4[j] = w1*(original_HE[j]*original_HE[j])+
w2*(meanIBD2[j]*meanIBD2[j]);

w1=0, w2=0, w3=0, w4=0;
if (-HE_DS[j]>0)   {w1=1;}
if (-IBD2_DS[j]>0) {w2=1;}
if (HE_CS[j]>0)    {w3=1;}
if (IBD2_CS[j]>0)  {w4=1;}
COMPOSITE_DAC4[j]=  w1*(HE_DS[j]*HE_DS[j]) +w2*(IBD2_DS[j]*IBD2_DS[j])
+w4*(IBD2_CS[j]*IBD2_CS[j]);
COMPOSITE_DAC5[j]= w1*(HE_DS[j]*HE_DS[j]) +w2*(IBD2_DS[j]*IBD2_DS[j]) +
w3*(HE_CS2[j]*HE_CS2[j]) +w4*(IBD2_CS[j]*IBD2_CS[j]);


//////////////////////////////////////
//output to file "statistics"       //
/* HEADING INCLUDED AS THE FIRST LINE*/
/////////////////////////////////////
fprintf(fp4,"%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf
%lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf %lf\n",
original_HE[j], new_HE[j], sum_HE[j], Forrest_HE[j], Xu_HE[j],
Viss_HE[j],Sham_HE[j],robust_Sham[j],score_standard[j],score_robust[j],score_
3[j],score_4[j],score_5[j],score_6[j], meanIBD1[j],meanIBD2[j],
meanIBD3[j],IBD1_EDAC[j], IBD2_EDAC[j], composite1[j],
composite2[j],composite3[j],
```

```
composite_DS4[j],composite_CS2[j],composite_CS3[j],
COMPOSITE_DAC11[j],COMPOSITE_DAC12[j],COMPOSITE_DAC2[j],COMPOSITE_DAC3[j],COM
POSITE_DAC6[j],COMPOSITE_DAC4[j],COMPOSITE_DAC5[j],LL_store[j]);


/*For checking*/
 total_n_DS += n_DS;
 total_n_CS += n_CS;


}//end of loop of N_samples

fclose(fp2);
fclose(fp4);
 fclose(fp9);
//fclose(user_parameter);

}/*end of else from right before loop over all N studies*/



////////////////////////////////////////////////////
/*calling power calculation subroutine from main() */
////////////////////////////////////////////////////
  power (LL_store, original_HE, sum_HE, new_HE, Xu_HE, Viss_HE,Forrest_HE,
       Sham_HE, robust_Sham, score_standard, score_robust, score_3,score_4,
       score_5, score_6, meanIBD1,meanIBD2,meanIBD3,
       composite1,composite2,composite3,
       composite_CS2, composite_CS3, COMPOSITE_DAC11, COMPOSITE_DAC12,
       COMPOSITE_DAC2,COMPOSITE_DAC3,
       composite_DS4, COMPOSITE_DAC4,COMPOSITE_DAC5,COMPOSITE_DAC6,
       IBD1_EDAC, IBD2_EDAC);


printf("Done!\n\n");
printf("\nPlease check file <user_parameters> for reminder of the sample
used\n");
printf("Please check file <powers> for power results\n");
//printf("Please use file <user_COMPOSITE> for R program (optimal
weights)\n");
/*for checking EDAC, the numbers should match that from simulation program*/
printf("Total_n_DS, Total_n_CS: %d\t%d\n\n",total_n_DS,total_n_CS);

free(trait1);/*deallocate memory from earlier*/
free(trait2);
free(est_IBD);

}/*end of main*/




/*************************************************************************/

////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////
```

```
//                                                                    //
// Regression subroutine for original_HE, sum_HE, new_HE, Sham ///
////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////

double regression(double * y, long n, double * x)
{

  if (n<3) return(0);

double  sum_xy = 0.0, sum_x = 0.0, sum_y = 0.0, sum_sqr_x = 0.0, RSS = 0.0;
double sigma_hat = 0.0;
int i=0;
double   SE = 0.0, beta = 0.0, value = 0.0;
double  intercept_est = 0.0;

for(i=0;i<n;i++){
                   sum_xy += y[i]*x[i];
                   sum_x  += x[i];
                   sum_y  += y[i];
                   sum_sqr_x += x[i]*x[i];
}

beta = (n*sum_xy - sum_x*sum_y)/(n*sum_sqr_x - sum_x*sum_x);
intercept_est = sum_y/n - beta*sum_x/n;

for(i=0;i<n;i++) {
         RSS += pow(y[i] - (intercept_est + beta*est_IBD[i]),2.0);
}
sigma_hat = sqrt(RSS/(n-2));
SE = sigma_hat/sqrt(sum_sqr_x - sum_x*sum_x/n );
value = beta/SE;

/* return standardized regression slope - the T-statistic*/
return (value);
}



////////////////////////////////////////////////////////
////////////////////////////////////////////////////////
//                                                    //
//                       Forrest HE                   //
//                                                    //
////////////////////////////////////////////////////////
////////////////////////////////////////////////////////

double Forrest(double * sum_HE,double *sqr_trait_diff,double *sqr_trait_sum,
long num_fam, double *est_IBD){

double mean_sqr_diff = 0.0, sum_y = 0.0, sum_x=0.0, sum_sqr_x = 0.0,
       sum_sqrsum_IBD = 0.0, sum_sqrdif_IBD = 0.0, mean_sqr_sum = 0.0,
       mean_sqr_IBD = 0.0, beta_temp = 0.0, junk = 0.0, alpha = 0.0,
       gama = 0.0;

int i = 0;
```

```
double sigma = 0.0, tau=0.0;  double mean_IBD = 0.0;
for(i=0;i<num_fam;i++){
                  mean_sqr_diff += sqr_trait_diff[i];
                  sum_y  += sqr_trait_sum[i];
                  sum_x += est_IBD[i];
                  sum_sqr_x += est_IBD[i]*est_IBD[i];
                  sum_sqrsum_IBD += sqr_trait_sum [i] * est_IBD[i];
                  sum_sqrdif_IBD += sqr_trait_diff[i] * est_IBD[i];
}

mean_sqr_diff/=num_fam; mean_sqr_sum = sum_y / num_fam; mean_IBD =sum_x
/num_fam;
mean_sqr_IBD = sum_sqr_x / num_fam;

beta_temp =  *sum_HE; junk = beta_temp+10.0;


while( fabs(beta_temp-junk) > .0000001 ){
       junk = beta_temp;
       alpha = mean_sqr_diff + beta_temp * mean_IBD;
       gama  = mean_sqr_sum  - beta_temp * mean_IBD;

       double RSS_sum = 0.0;double  RSS_sum2 = 0.0;
      for( i = 0; i < num_fam; i++){

      RSS_sum += pow( sqr_trait_diff[i] - alpha + beta_temp*est_IBD[i], 2.0);
      RSS_sum2+= pow( sqr_trait_sum[i] - gama  - beta_temp*est_IBD[i], 2.0);
      }
      sigma = sqrt( RSS_sum/num_fam );
      tau   = sqrt( RSS_sum2/num_fam);


      /*the weighted beta*/
       beta_temp = (pow(sigma/tau,2.0) * (sum_sqrsum_IBD - gama *
                   num_fam*mean_IBD )
                  -(sum_sqrdif_IBD - alpha* num_fam*mean_IBD ))/
                   (  (1+pow(sigma/tau,2.0))*num_fam*mean_sqr_IBD );

  }

double beta_wtHE = beta_temp;  /*assign the betas to the array*/

/*standard error for the weighted beta*/
double SE_beta_wtHE = sqrt(1.0/(num_fam*mean_sqr_IBD)) *
                  sqrt( 1.0/ ( (1.0/(tau*tau)) + (1.0/(sigma*sigma)) ) );

double Forrest_HE = 0.0;
Forrest_HE = beta_wtHE/SE_beta_wtHE;

/* return standardized Forrest weighted H-E slope(T-statistic)*/
return(Forrest_HE);

}
```

```
////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////
//
//Xu regression - self contained trait sum and trait diff regressions//
//This also calculates Visscher&Hopper weighted HE                    //
//                                                                    //
////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////


/*********************************************************************/
/*1)Regression subroutine are done twice. Once for sum, once for     */
/*difference. y[i] take on two values: sqr_trait_sum & sqr_trait_diff*/
/*2)Use un-normalized IBD as independent variable as in Xu's paper   */
/*3)Names such as beta1, beta2 are consistent with Xu's paper        */
/*4)Xu's HE allows covariance of beta1 and beta2, but not in Visscher*/
/*and Hopper's version. Otherwise they are the same.                 */

/*********************************************************************/

double Xu(double *sqr_trait_sum, double *sqr_trait_diff, long num_fam,
          double *IBD, double *Viss_HE) {

int i = 0;

////////////////////////////////////////////////////
//Regress sqr_trait_sum on IBD for beta2 & SE(beta2)//
////////////////////////////////////////////////////
  double sum_xy = 0.0, sum_y = 0.0, sum_x = 0.0, sum_beta =
0.0,sum_intercept_est = 0.0,  sum_sqr_x = 0.0, RSS = 0.0;
  double sum_sigma_hat = 0.0, sum_Value = 0.0, sum_SE=0.0;
  double sum_intercept = 0.0;

for(i=0;i<num_fam;i++){
                  sum_xy += sqr_trait_sum[i]*IBD[i];
                  sum_x  += IBD[i];
                  sum_y  += sqr_trait_sum[i] ;
                  sum_sqr_x +=IBD[i]*IBD[i];
}
sum_beta = (num_fam*sum_xy - sum_x*sum_y)/(num_fam*sum_sqr_x - sum_x*sum_x);
sum_intercept_est = sum_y/num_fam - sum_beta*sum_x/num_fam;

/* calculation for SE(beta2)*/
for(i=0;i<num_fam;i++) {
RSS += pow(sqr_trait_sum[i] - (sum_intercept_est + sum_beta*IBD[i]),2.0);
}

sum_sigma_hat = sqrt(RSS/(num_fam-2));
sum_SE = sum_sigma_hat/sqrt(sum_sqr_x - sum_x*sum_x/num_fam );


////////////////////////////////////////////////////
//Regress sqr_trait_diff on IBD for [-beta1] & SE(beta1)//
////////////////////////////////////////////////////

/*define/reset the following to zero for the new regression*/
/*sum_x and sum_sqr_x are the same as previous*/
```

```
  sum_xy = 0.0, sum_y = 0.0,  RSS = 0.0;
  double diff_sigma_hat = 0.0, diff_beta = 0.0, diff_Value = 0.0;
  double var_beta_diff = 0.0, var_beta_sum = 0.0, weight = 0.0,
cov_beta_diff_beta_sum = 0.0, diff_SE =0.0, Xu_HE=0.0;
  double diff_intercept_est = 0.0;

for(i=0;i<num_fam;i++){
                sum_xy += sqr_trait_diff[i]*IBD[i];
                sum_y  += sqr_trait_diff[i] ;
}
diff_beta = (num_fam*sum_xy - sum_x*sum_y)/(num_fam*sum_sqr_x - sum_x*sum_x);
diff_intercept_est = sum_y/num_fam - diff_beta*sum_x/num_fam;

/*by regression formulae, diff_beta is the slope so with sign*/
//printf("inside Xu subroutine: diff_slope   diff_intercept_est %lf %lf\n",
diff_beta, diff_intercept_est);


/*calculation for SE(beta1)*/
for(i=0;i<num_fam;i++) {
RSS += pow(sqr_trait_diff[i] - (diff_intercept_est +diff_beta*IBD[i]),2.0);
}
diff_sigma_hat = sqrt(RSS/(num_fam-2));
diff_SE = diff_sigma_hat/sqrt(sum_sqr_x - sum_x*sum_x/num_fam );


////////////////////////////////////
/*calculation for COV(beta1,beta2)*/
////////////////////////////////////
double * est_trait_sum = new double [num_fam];
double * est_trait_diff = new double [num_fam];
double * error_trait_sum = new double[num_fam];
double * error_trait_diff  = new double[num_fam];
double cov_errors = 0.0;

/* COV(beta1,beta2)'s numerator is the COV(error_sum,error_diff)*/
/*error_sum&erro_diff are estimated errors from each regression */
/*e.g.,error_diff=sqrt_trait_diff- [est_alpha1 - est_beta1*IBD]*/

for(i=0; i<num_fam; i++)  {
    est_trait_sum[i] = sum_intercept_est + sum_beta * IBD[i];
    est_trait_diff[i] = diff_intercept_est + diff_beta *IBD[i];
    error_trait_sum[i] = sqr_trait_sum[i] - est_trait_sum[i];
    error_trait_diff[i] = sqr_trait_diff[i] - est_trait_diff[i];
}

for (i=0; i<num_fam; i++){
cov_errors += error_trait_diff[i] *error_trait_sum[i]/num_fam;
}

/* COV(beta1,beta2)'s denominator is sum_IBD_dev_sq*/
double meanIBD = 0.0;
for (i = 0; i< num_fam; i++) {meanIBD +=IBD[i];}
/*meanIBD is the average un_normalized IBD for the study*/
meanIBD /= (double)num_fam;

double sum_IBD_dev_sq = 0.0;
```

```
for(i=0;i<num_fam;i++){
    sum_IBD_dev_sq += (IBD[i]-meanIBD)* (IBD[i]-meanIBD);
}

cov_beta_diff_beta_sum =-cov_errors / sum_IBD_dev_sq;
//printf("inside Xu subroutine: covbeta_diff_beta_sum: %lf\n",
cov_beta_diff_beta_sum);



////////////////////////////////////////////////////////////
/*use COV(beta1,beta2), VAR(beta1),VAR(beta2)to get weights*/
/*and weighted overall beta and its variance (overall_beta)*/
////////////////////////////////////////////////////////////
var_beta_diff = diff_SE* diff_SE;
var_beta_sum = sum_SE * sum_SE;
weight = (var_beta_sum - (cov_beta_diff_beta_sum)) / (var_beta_diff +
var_beta_sum - (2 * cov_beta_diff_beta_sum));




////////////////////////////////////////////////////////////
//Xu_HE is the standardized Xu weighted H-E slope(T-statistic)//
////////////////////////////////////////////////////////////

/*weighted Xu_HE = (weight * beta1) + ((1 - weight) * beta2)*/
double overall_beta = 0.0;
diff_beta = 0.0-diff_beta;
/*diff_beta is the slope, -diff_beta is the beta1 in Xu's paper*/
overall_beta = (weight * diff_beta) + ((1 - weight) * sum_beta);

double SE_overall_beta = 0.0;
SE_overall_beta = sqrt(  (var_beta_diff *var_beta_sum -
         cov_beta_diff_beta_sum *cov_beta_diff_beta_sum)/
(var_beta_diff + var_beta_sum - 2 * cov_beta_diff_beta_sum)  );


Xu_HE= overall_beta/SE_overall_beta;




////////////////////////////////////////////////////////////
/*Visscher and Hopper use VAR(beta1),VAR(beta2)to get weights*/
/*weighted overall beta and its variance (overall_beta)*/
////////////////////////////////////////////////////////////
 double Viss_weight = 0.0;
 Viss_weight = var_beta_sum/(var_beta_diff + var_beta_sum );

 /*weighted Viss_HE = (weight * beta1) + ((1 - weight) * beta2)*/
 double Viss_weighted_beta = 0.0;
 /*diff_beta and sum_beta are the same as Xu's HE*/
 Viss_weighted_beta = (weight * diff_beta) + ((1 - weight) * sum_beta);

 double SE_Viss_beta = 0.0;
```

```
 SE_Viss_beta = sqrt(  (var_beta_diff *var_beta_sum)/
                   (var_beta_diff + var_beta_sum)  );

 ////////////////////////////////////////////////////////////////////
 //Viss_HE is the standardized Visscher & Hopper weighted H-E(T-stat)//
 ////////////////////////////////////////////////////////////////////

 * Viss_HE= Viss_weighted_beta/SE_Viss_beta;


/* return the standardized Xu_HE to main(), NOTE Viss_HE is a pointer
variable*/
return(Xu_HE);

}



////////////////////////////////////////////////////////////////////////////
/
////////////////////////////////////////////////////////////////////////////
//
//
//
//Tang and Siegmunds' score statistics, use the values score_c[i]
//
// (Ai/4 function)
//
//
//
//Tang1 = SCORE1,the standard score statistic, using fisher information
//
//Tang2 = SCORE2, using  partically empirical variance of the numerator
//
//Tang3 = SCORE3, using  fully empirical variance of the numerator
//
//Tang4 = SCORE4, using  pi-bar in both numerater and demoninator
//
//Tang5 = SCORE5, using  pi-bar in denominator, one half in numerator
//
//Tang6 = SCORE6, using sqr_trait_diff instead of outcome_robust_sham
//
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////


void Tang(double *score_c,double *score_D, long num_fam,double *est_IBD,
double corr,
         double *Tang1 ,double *Tang2,double *Tang3,double *Tang4, double
*Tang5, double *Tang6)

    {
        double sum_sq_ci = 0.0;
        double sum_sq_Di = 0.0;
      double numerator=0.0;
      double denominator_standard=0.0 ;
      double denominator_2=0.0;
```

```
      double numerator_4=0.0;
      double denominator_3=0.0 ;
      double denominator_4=0.0;
        double denominator_6=0.0;
        double numerator_6=0.0;
        long i=0;

      /* for Tang4=score_4*/
      double meanIBD = 0.0;
      for (i = 0; i< num_fam; i++) {meanIBD +=est_IBD[i];}
      /*meanIBD is the average normalized IBD for the study*/
      meanIBD /= (double)num_fam;

      double sum_IBD_dev_half = 0.0;
        double sum_IBD_dev_meanIBD= 0.0;
      for(i=0;i<num_fam;i++){
        sum_IBD_dev_half += est_IBD[i]*est_IBD[i];
        sum_IBD_dev_meanIBD += (est_IBD[i]-meanIBD)* (est_IBD[i]-meanIBD);
      }


        for(i=0;i<num_fam;i++) {
        numerator += (2.0*est_IBD[i])*score_c[i];
          numerator_4 += (est_IBD[i]-meanIBD)*score_c[i];
          numerator_6 += (2.0*est_IBD[i])*score_D[i];
          sum_sq_ci +=score_c[i]*score_c[i];
          sum_sq_Di +=score_D[i]*score_D[i];
            }


      denominator_standard= sqrt(num_fam*(1+corr*corr)/((1-corr*corr)*
                                                (1-corr*corr))/2);
        denominator_2 = sqrt(sum_sq_ci/2);
        denominator_3 = sqrt(sum_sq_ci*sum_IBD_dev_half/num_fam);
        denominator_4 = sqrt(sum_sq_ci*sum_IBD_dev_meanIBD/num_fam);
        denominator_6 = sqrt(sum_sq_Di*sum_IBD_dev_half/num_fam);

        * Tang1=numerator/denominator_standard;
      * Tang2=numerator/denominator_2;
      * Tang3=(0.5*numerator)/denominator_3;
      * Tang4=numerator_4/denominator_4;
        * Tang5=(0.5*numerator)/denominator_4;
        * Tang6=(0.5*numerator_6)/denominator_6;

}




////////////////////////////////////////////////////////
////////////////////////////////////////////////////////
//                                                    //
// robust Sham regression                             //
//        - fixed intercept & Ai as depedent variable //
//                                                    //
////////////////////////////////////////////////////////
```

```
//////////////////////////////////////////////////////

/* regress outcome_robustSham on est_IBD with no intercept*/

double robustShamreg(double * outcome_robustSham, long num_fam, double *
est_IBD){

double top_beta = 0.0, bottom_beta = 0.0, beta_Sham = 0.0, sum_sqr_y = 0.0,
sum_x = 0.0, variance_robustSham = 0.0,
top_SE = 0.0, std_err_robust_Sham ;
int i=0;

for(i=0;i<num_fam;i++){
   top_beta += (est_IBD[i]  *  outcome_robustSham[i]);
   bottom_beta += (est_IBD[i]  * est_IBD[i]);
}

beta_Sham = top_beta/bottom_beta;

for(i=0; i<num_fam;i++){
     top_SE += (outcome_robustSham[i] - (beta_Sham * est_IBD[i])) *
            (outcome_robustSham[i] - (beta_Sham * est_IBD[i])) ;
}
std_err_robust_Sham = sqrt (top_SE/(num_fam * bottom_beta));

/*standardized robust_Sham H-E slope(T-statistic) is the output*/
double robust_Sham = 0.0;
robust_Sham = beta_Sham/std_err_robust_Sham;
return(robust_Sham);

}







//////////////////////////////////////////////////////////////////////////
//
//////////////////////////////////////////////////////////////////////////
//
//
/
//              Variance components from  Bill Forrest's  original code
/
//
/
//////////////////////////////////////////////////////////////////////////
//
//////////////////////////////////////////////////////////////////////////
/

double Vc(double *trait1, double *trait2, double *est_IBD, double
trait_mean, double trait_var, double trait_cov, long num_fam){
```

```
  ////////////////////////////////////////////////////////////////
  // Now we maximize the loglikelihood under the null and alternative.
  // Note that the double-precision variable "trait_mean" already
  // contains the MLE (when genetic variance is 0) of the common mean mu.
  ////////////////////////////////////////////////////////////////

  int i=0;

  double lambda = 0.0;    // transformed correlation value.
                          // correlation = (exp(lambda)-1)/(exp(lambda)+1)


  lambda = log( (1.0 + trait_cov/trait_var) / (1.0 - trait_cov/trait_var) );
// initial value
  sig_e = sqrt(trait_var); // initial value
  mu = trait_mean;   // initial value
  sig_g = 0.0; // null-hypothesis value.

     double * LL_val_res = new double[4];
     long nfunk = 0, pinit1, pinit2;
     double **p3 = new double*[4];
     double parval3[3] = {mu, sig_e, lambda};
     for( pinit1 = 0; pinit1 <  4; pinit1++){
        p3[pinit1] = new double[3];
        for( pinit2 = 0; pinit2 <  3; pinit2++)
            p3[pinit1][pinit2] = parval3[pinit2] + 0.1*rnorm();
        LL_val_res[pinit1] = LL_res(3,p3[pinit1]);

     }

  simplex( LL_res, p3, LL_val_res, 3, .00001, &nfunk);
  mu = p3[0][0]; sig_e = p3[0][1]; lambda = p3[0][2];
  parval3[0] = mu; parval3[1] = sig_e; parval3[2] = lambda;


  //est_IBD fed in from main section
  double LL_null = loglike(mu, sig_e, lambda, 0.0, trait1, trait2,
est_IBD,num_fam);



 for( pinit1 = 0; pinit1 <  4; pinit1++){
        p3[pinit1] = new double[3];
        if(pinit1 > 0){
            if(pinit1==1)
            {p3[1][0] = parval3[0] + 0.02; p3[1][1] = parval3[1]; p3[1][2] =
parval3[2];}
            if(pinit1==2)
            {p3[2][0] = parval3[0]; p3[2][1] = parval3[1]+.02; p3[2][2] =
parval3[2];}
            if(pinit1==3)
            {p3[3][0] = parval3[0]; p3[3][1] = parval3[1]; p3[3][2] =
parval3[2]+.025;}
        }
        LL_val_res[pinit1] = K3(3,p3[pinit1]);
     }
```

```
simplex( K3, p3, LL_val_res, 3, .00001, &nfunk); //call to simplex.h

  mu = p3[0][0]; sig_e = p3[0][1]; lambda = p3[0][2];
  parval3[0] = mu; parval3[1] = sig_e; parval3[2] = lambda;

LL_null = loglike(mu, sig_e, lambda, 0.0, trait1, trait2, est_IBD,num_fam);


  // Now, we do the likelihood again, but this time with a component
  // of genetic variance scaled by estimated IBD sharing.


     double  sig_g = sig_e/2.0;

     double mu_old = mu, sig_e_old = sig_e, lambda_old = lambda, sig_g_old =
sig_g;
     double parval4[4] = {mu, sig_e, lambda, sig_g};

     double * LL_val4 = new double[4+1];
     double **p4 = new double*[4+1];

     for( pinit1 = 0; pinit1 <  (4+1); pinit1++){
        p4[pinit1] = new double[4];
       {
            if(pinit1==0)
              {p4[0][0] = parval4[0];       p4[0][1] = parval4[1];
               p4[0][2] = parval4[2];       p4[0][3] = parval4[3];}
            if(pinit1==1)
            {p4[1][0] = parval4[0] + 0.1; p4[1][1] = parval4[1];
               p4[1][2] = parval4[2];       p4[1][3] = parval4[3];}
            if(pinit1==2)
            {p4[2][0] = parval4[0]; p4[2][1] = parval4[1] + 0.1;
               p4[2][2] = parval4[2]; p4[2][3] = parval4[3];}
            if(pinit1==3)
            {p4[3][0] = parval4[0];      p4[3][1] = parval4[1];
               p4[3][2] = parval4[2]+0.1; p4[3][3] = parval4[3];}
            if(pinit1==4)
            {p4[4][0] = parval4[0];      p4[4][1] = parval4[1];
               p4[4][2] = parval4[2];      p4[4][3] = parval4[3]+0.1;}
        }
        LL_val4[pinit1] = LL_full(4,p4[pinit1]);
     }

simplex( LL_full, p4, LL_val4, 4, .00001, &nfunk);

// Store all the results:
mu = p4[0][0]; sig_e = p4[0][1]; lambda = p4[0][2]; sig_g = p4[0][3];
parval4[0] = mu; parval4[1] = sig_e; parval4[2] = lambda; parval4[3] = sig_g;


// Now we fine-tune the point estimate with the L2-gradient function.

     for( pinit1 = 0; pinit1 <  (4+1); pinit1++){

       {
            if(pinit1==0)
              {p4[0][0] = parval4[0];       p4[0][1] = parval4[1];
```

```
            p4[0][2] = parval4[2];       p4[0][3] = parval4[3];}
         if(pinit1==1)
         {p4[1][0] = parval4[0] + 0.1; p4[1][1] = parval4[1];
            p4[1][2] = parval4[2];       p4[1][3] = parval4[3];}
         if(pinit1==2)
         {p4[2][0] = parval4[0]; p4[2][1] = parval4[1] + 0.1;
            p4[2][2] = parval4[2]; p4[2][3] = parval4[3];}
         if(pinit1==3)
         {p4[3][0] = parval4[0];      p4[3][1] = parval4[1];
            p4[3][2] = parval4[2]+0.1; p4[3][3] = parval4[3];}
         if(pinit1==4)
         {p4[4][0] = parval4[0];      p4[4][1] = parval4[1];
            p4[4][2] = parval4[2];      p4[4][3] = parval4[3]+0.1;}
      }
      LL_val4[pinit1] = K4(4,p4[pinit1]);
   }


simplex( K4, p4, LL_val4, 4, .00001, &nfunk);

// Store all the results:
mu = p4[0][0]; sig_e = p4[0][1]; lambda = p4[0][2]; sig_g = p4[0][3];
parval4[0] = mu; parval4[1] = sig_e; parval4[2] = lambda; parval4[3] = sig_g;


double LL_alt = loglike(mu, sig_e, lambda, sig_g, trait1, trait2,
est_IBD,num_fam);


printf("\nLL_null,LL_alt,2.0*(LL_alt-LL_null)\n");
printf("%lf %lf %lf\n",LL_null,LL_alt,2.0*(LL_alt-LL_null));



//LL_store = 2.0*(LL_alt-LL_null);

return(2.0*(LL_alt-LL_null));
}//end of subroutine for Vc



////////////////////////////////////////////////////////////////////
// SUPPORT FUNCTION DEFINITIONS for variance components
////////////////////////////////////////////////////////////////////

double LL_res (int n_val, double * XPAR){ // restrict genetic variance to be
0.0.
   return( -loglike( XPAR[0], XPAR[1], XPAR[2], 0.0, trait1, trait2, est_IBD,
num_fam) );
}

double LL_full(int n_val, double *XPAR){
   return( -loglike( XPAR[0], XPAR[1], XPAR[2], XPAR[3], trait1, trait2,
est_IBD, num_fam) );
}
```

```
double K3(int n_val, double *x){

     double mu = x[0]; double sig_e = fabs(x[1]); double lambda = x[2];
double sig_g = 0.0;

     double L2grad = 0.0, temp_mu = 0.0, temp_sig_e = 0.0, temp_lambda = 0.0;

     double correlation = (exp(lambda)-1.0)/(exp(lambda)+1.0);

     for (int i = 0; i < num_fam; i++){



        double Yd = (trait1[i] - trait2[i])/sqrt(2.0);
        double Ys = (trait1[i] - mu + trait2[i] - mu)/sqrt(2.0);
      // This diagonalizes the data, so we can work with
        // two univariate normals instead of one bivariate.

        double hd = sig_e*sig_e*(1.0-correlation);  // sig_g == 0
        double hs = sig_e*sig_e*(1.0+correlation);  // sig_g == 0

        temp_mu     += sqrt(2.0) * Ys/hs;

        temp_sig_e  += (-0.5/hd + 0.5*Yd*Yd/(hd*hd)) * 2*sig_e*(1-
correlation) +
                       (-0.5/hs + 0.5*Ys*Ys/(hs*hs)) *
2*sig_e*(1+correlation);

        temp_lambda += (-0.5/hd + 0.5*Yd*Yd/(hd*hd)) * (-sig_e*sig_e) *
                           2*exp(lambda)/pow(exp(lambda)+1.0,2.0)+
                       (-0.5/hs + 0.5*Ys*Ys/(hs*hs)) *  (sig_e*sig_e) *
                           2*exp(lambda)/pow(exp(lambda)+1.0,2.0);
    }

   L2grad = (sqrt( pow(temp_mu,2.0) + pow(temp_sig_e,2.0) +
pow(temp_lambda,2.0) ));

   if(0){
   cout << mu << "  " << sig_e << "  " <<  lambda << "  "
        << loglike(mu, sig_e, lambda, 0.0, trait1, trait2, est_IBD, num_fam)
<< endl;
   cout << " L2 of restricted gradient is " << L2grad << endl;
   }

   return L2grad;

  }




double K4(int n_val, double *x){


     double mu = x[0];
     double sig_e = fabs(x[1]);
     double lambda = x[2];
```

```
     double sig_g = fabs(x[3]);

     double L2grad = 0.0, temp_mu = 0.0, temp_sig_e = 0.0, temp_lambda = 0.0,
temp_sig_g = 0.0;

     double correlation     = (exp(lambda)-1.0)/(exp(lambda)+1.0);
     double d_correlation_d_lambda = 2*exp(lambda)/pow(exp(lambda)+1.0,2.0);

     for (int i = 0; i < num_fam; i++){

        double Yd = (trait1[i] - trait2[i])/sqrt(2.0);
        double Ys = (trait1[i] - mu + trait2[i] - mu)/sqrt(2.0);
      // This diagonalizes the data, so we can work with
        // two univariate normals instead of one bivariate.

        double hd = sig_e*sig_e*(1.0-correlation) + sig_g*sig_g*(1.0-
est_IBD[i]);
        double hs = sig_e*sig_e*(1.0+correlation) +
sig_g*sig_g*(1.0+est_IBD[i]);

        temp_mu     += sqrt(2.0) * Ys/hs;

        temp_sig_e  += (-0.5/hd + 0.5*Yd*Yd/(hd*hd)) * 2*sig_e*(1-
correlation) +
                       (-0.5/hs + 0.5*Ys*Ys/(hs*hs)) *
2*sig_e*(1+correlation);

        temp_lambda += (-0.5/hd + 0.5*Yd*Yd/(hd*hd)) * (-sig_e*sig_e) *
                           d_correlation_d_lambda       +
                       (-0.5/hs + 0.5*Ys*Ys/(hs*hs)) *  (sig_e*sig_e) *
                           d_correlation_d_lambda;

        temp_sig_g +=  (-0.5/hd + 0.5*Yd*Yd/(hd*hd)) * 2*sig_g*(1.0 -
est_IBD[i]) +
                       (-0.5/hs + 0.5*Ys*Ys/(hs*hs)) * 2*sig_g*(1.0 +
est_IBD[i]);
    }

   L2grad = sqrt( pow(temp_mu,2.0) + pow(temp_sig_e,2.0) +
pow(temp_lambda,2.0) + pow(temp_sig_g,2.0));

   if(0){
   cout << "Point = ( " << mu << "  " << sig_e << "  " <<  lambda
       << "  " << sig_g << ") --> " << L2grad << "   LL = "
       << loglike(mu,sig_e,lambda,sig_g,trait1,trait2,est_IBD,num_fam) <<
endl;
   cout << "Gradient value = ( " << temp_mu << " " << temp_sig_e << " "
       << temp_lambda << " " << temp_sig_g << endl;
      cout << " L2 of restricted gradient is " << L2grad << endl;
   }

   return L2grad;

  }
```

```
double K4b(int n_val, double *x){
 // A numerical version of K4().

     double mu = x[0]; double sig_e = x[1]; double lambda = x[2]; double
sig_g = x[3];


double little = 0.0000001;
double L2_grad = sqrt(pow((loglike(mu+little,sig_e,lambda,
sig_g,trait1,trait2,est_IBD,num_fam)-loglike(mu,sig_e,lambda,
sig_g,trait1,trait2,est_IBD,num_fam))/little,2.0)
+pow((loglike(mu,sig_e+little,lambda, sig_g,trait1,trait2,est_IBD,num_fam)-
loglike(mu,sig_e,lambda, sig_g,trait1,trait2,est_IBD,num_fam))/little,2.0) +
pow((loglike(mu,sig_e,lambda+little, sig_g,trait1,trait2,est_IBD,num_fam)-
loglike(mu,sig_e,lambda, sig_g,trait1,trait2,est_IBD,num_fam))/little,2.0) +
pow((loglike(mu,sig_e,lambda, sig_g+little,trait1,trait2,est_IBD,num_fam)-
loglike(mu,sig_e,lambda, sig_g,trait1,trait2,est_IBD,num_fam))/little,2.0));

cout << mu << "  " << sig_e << "  " <<  lambda << "  " << sig_g << "    " <<
L2_grad << endl;


     return L2_grad;
}
```

```
double loglike( double MU, double SIG_E, double LAMBDA, double SIG_G,
        const double * trait1, const double * trait2, const double * est_IBD,
                long num_fam){

  // MU = parameter for common bivariate mean
  // SIG_E = sqrt of individual environmental variance
  // LAMBDA = log( (1.0+correlation) / (1.0-correlation) )
  // SIG_G = sqrt of genetic variance (additive + dominance)
  // * trait1 = pointer to sib 1 trait data
  // * trait2 = pointer to sib 2 trait data
  // * est_IBD = estimated IBD sharing at marker locus
  // num_fam = length of data (i.e. of trait1, trait2, and est_IBD)

    double out = 0.0;
    double correlation = (exp(LAMBDA)-1.0)/(exp(LAMBDA)+1.0);

    for (int i = 0; i < num_fam; i++){

        double Yd = (trait1[i] - trait2[i])/sqrt(2.0);
        double Ys = (trait1[i] + trait2[i])/sqrt(2.0) - sqrt(2.0)*MU;
      // This diagonalizes the data, so we can work with
        // two univariate normals instead of one bivariate.
```

```
        double hd = SIG_E*SIG_E*(1.0-correlation) + SIG_G*SIG_G*(1.0-
est_IBD[i]);
        double hs = SIG_E*SIG_E*(1.0+correlation) +
SIG_G*SIG_G*(1.0+est_IBD[i]);

        out += (- 0.50*log(hd) - 0.50*Yd*Yd/hd);
        out += (- 0.50*log(hs) - 0.50*Ys*Ys/hs);
  }

  out -= num_fam*log(2*(3.141592653));  // a factor of -n*log(2*Pi), for
completeness

  return(out);
}




////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////
//                                                                    //
//             Power calculations for all the statistics above        //
//                   output is the file "powers"                      //
//                                                                   //
//All the tests are 1-sided tests although they're not all the same  //
//side.                                                              //
// For  meanIBD statistics, we currently                             //
//compute two  versions of the power for it, one for each side.      //
//Then in any given run of the program we can just use whichever     //
//of those we know to be the correct one in any particular           //
// situation.                                                        //
//                                                                    //
//N=200 is large enough to treat t's as z's, so we use Z cutoff for  //
//everything but variance components.                                //
//The cutoff for 0.01 significance for the Z(1-sided) is 2.326       //
// The cutoff for 0.01 significance for the chi-squared is 6.63.     //
//These are hard coded here.                                         //
//                                                                    //
//variance components:chi-square 1 df                                //
//original HE: t (n - 2) 1-sided (negative)                          //
//trait sum: t (n - 2) 1-sided (positive)                            //
//new HE: t (n - 2) 1-sided (positive)                               //
//Xu: asymptotically Z 1-sided (negative)                            //
//Forrest's method: asymptotically Z 1-sided (positive)              //
//Sham (non-robust): t (n - 2) 1-sided (positive)                    //
//Sham (robust): t (n - 1) 1-sided (positive)                        //
//all the 6 SCORE statistics: asymptotically Z 1-sided (positive)    //
//IBD-sharing statistics:  asymptotically Z 1-sided                  //
//             - for discordant pairs: negative side                 //
//             - for concordant pairs: positive side                 //
//             - for EDAC sib pairs: positive side                   //
//COMPOSITE statistics: asymptotically Z 1-sided (depends)           //
//             - for discordant pairs: negative side                 //
//             - for concordant pairs: positive side                 //
//             - for EDAC sib pairs: positive side                   //
//                                                                    //
```

```
//For the maximized composite statistics, cutoffs are found empirically//
//based on their mixture chi-sq distribtuions.                          //
////////////////////////////////////////////////////////////////////////


void power (double *LL_store, double *original_HE, double *sum_HE,
          double *new_HE, double *Xu_HE, double *Viss_HE,double *Forrest_HE,
            double *Sham_HE, double *robust_Sham,
            double *score_standard, double *score_robust,
            double *score_3,double *score_4,
            double *score_5,double *score_6,
            double *meanIBD1, double *meanIBD2, double *meanIBD3,
            double *composite1, double *composite2, double *composite3,
            double *composite_CS2, double *composite_CS3,
            double *COMPOSITE_DAC11, double *COMPOSITE_DAC12,
            double *COMPOSITE_DAC2, double *COMPOSITE_DAC3,
            double *composite_DS4, double *COMPOSITE_DAC4,
            double *COMPOSITE_DAC5,double *COMPOSITE_DAC6,
            double *IBD1_EDAC, double *IBD2_EDAC)

{


int j=0; /* counter for loop N-samples*/

FILE *fp5; /*file pointer for outputted power*/
fp5=fopen("powers", "w");


  /* Declare and initialize variables for */
  /* the power of each test*/

  double power_VC = 0.0 ;
  double power_original_HE = 0.0 ;
  double power_sum_HE = 0.0 ;
  double power_new_HE = 0.0 ;
  double power_Xu_HE = 0.0 ;
  double power_Viss_HE = 0.0 ;
  double power_Forrest_HE = 0.0 ;
  double power_Sham_HE = 0.0 ;
  double power_robust_Sham = 0.0 ;
  double power_score_standard_pos = 0.0 ;
  double power_score_robust_pos = 0.0 ;
  double power_score_3 = 0.0 ;
  double power_score_4 = 0.0 ;
  double power_score_5 = 0.0 ;
  double power_score_6 = 0.0 ;
  double power_meanIBD1_pos = 0.0 ;
  double power_meanIBD1_neg = 0.0 ;
  double power_meanIBD2_pos = 0.0 ;
  double power_meanIBD2_neg = 0.0 ;
  double power_meanIBD3_pos = 0.0 ;
  double power_meanIBD3_neg = 0.0 ;
  double power_composite1 = 0.0 ;
  double power_composite2 = 0.0 ;
  double power_composite3 = 0.0 ;
  double power_composite_CS2 = 0.0 ;
```

```
double power_composite_CS3 = 0.0 ;
double power_COMPOSITE_DAC11 = 0.0 ;
double power_COMPOSITE_DAC12 = 0.0 ;
double power_COMPOSITE_DAC2 = 0.0 ;
double power_COMPOSITE_DAC3 = 0.0 ;
double power_COMPOSITE_DAC6 = 0.0 ;
double power_IBD1_EDAC = 0.0 ;
double power_IBD2_EDAC = 0.0 ;
double power_composite_DS4 = 0.0 ;
double power_COMPOSITE_DAC4 = 0.0 ;
double power_COMPOSITE_DAC5 = 0.0 ;



/* Declare and initialize variables for the */
/* average value of each test statistics */
/* unsquared Z and ch-squared */

double ave_VC = 0.0 ;
double ave_original_HE = 0.0 ;
double ave_sum_HE = 0.0 ;
double ave_new_HE = 0.0 ;
double ave_Xu_HE = 0.0 ;
double ave_Viss_HE = 0.0 ;
double ave_Forrest_HE = 0.0 ;
double ave_Sham_HE = 0.0 ;
double ave_robust_Sham = 0.0 ;
double ave_score_standard = 0.0 ;
double ave_score_robust = 0.0 ;
double ave_score_3 = 0.0 ;
double ave_score_4 = 0.0 ;
double ave_score_5 = 0.0 ;
double ave_score_6 = 0.0 ;
double ave_meanIBD1 = 0.0 ;
double ave_meanIBD2 = 0.0 ;
double ave_meanIBD3 = 0.0 ;
double ave_composite1 = 0.0 ;
double ave_composite2 = 0.0 ;
double ave_composite3 = 0.0 ;
double ave_composite_CS2 = 0.0 ;
double ave_composite_CS3 = 0.0 ;
double ave_COMPOSITE_DAC11 = 0.0 ;
double ave_COMPOSITE_DAC12 = 0.0 ;
double ave_COMPOSITE_DAC2 = 0.0 ;
double ave_COMPOSITE_DAC3 = 0.0 ;
double ave_COMPOSITE_DAC6 = 0.0 ;
double ave_IBD1_EDAC = 0.0 ;
double ave_IBD2_EDAC = 0.0 ;
double ave_composite_DS4 = 0.0 ;
double ave_COMPOSITE_DAC4 = 0.0 ;
double ave_COMPOSITE_DAC5 = 0.0 ;


/*for power calculation of each test*/
for ( j = 0; j < N_samples; j++) {

  if  (LL_store[j] > 6.63)  power_VC++;
```

```
    if  (original_HE[j] < -2.326 )  power_original_HE++;
    if  (sum_HE[j] >2.326)  power_sum_HE++;
    if  (new_HE[j] > 2.326)  power_new_HE++;
    if  (Xu_HE[j] > 2.326)  power_Xu_HE++;
    if  (Viss_HE[j] > 2.326)  power_Viss_HE++;
    if  (Forrest_HE[j] > 2.326)  power_Forrest_HE++;
    if  (Sham_HE[j] > 2.326)  power_Sham_HE++;
    if  (robust_Sham[j] >2.326)  power_robust_Sham++;

    if  (score_standard[j] >2.326)  power_score_standard_pos++;
    if  (score_robust[j] >2.326)  power_score_robust_pos++;
    if  (score_3[j] >2.326)  power_score_3++;
    if  (score_4[j] >2.326)  power_score_4++;
    if  (score_5[j] >2.326)  power_score_5++;
    if  (score_6[j] <-2.326)  power_score_6++;

    if  (meanIBD1[j] > 2.326)  power_meanIBD1_pos++;
    if  (meanIBD1[j] < -2.326)  power_meanIBD1_neg++;
    if  (meanIBD2[j] > 2.326)  power_meanIBD2_pos++;
    if  (meanIBD2[j] < -2.326)  power_meanIBD2_neg++;
    if  (meanIBD2[j] > 2.326)  power_meanIBD3_pos++;
    if  (meanIBD3[j] < -2.326)  power_meanIBD3_neg++;

    /*for MDSP, composite statistic is negative*/
    if  (composite1[j] < -2.326)  power_composite1++;
    if  (composite2[j] < -2.326)  power_composite2++;
    if  (composite3[j] < -2.326)  power_composite3++;


    /*for concordant pairs, composite statistic is positive*/
    if  (composite_CS2[j] > 2.326)  power_composite_CS2++;
    if  (composite_CS3[j] > 2.326)  power_composite_CS3++;

    /*The combined discordant and concordant pairs*/
    if  (COMPOSITE_DAC11[j] > 2.326)  power_COMPOSITE_DAC11++;
    if  (COMPOSITE_DAC12[j] < -2.326)  power_COMPOSITE_DAC12++;
    if  (COMPOSITE_DAC2[j] > 2.326)  power_COMPOSITE_DAC2++;
    if  (COMPOSITE_DAC3[j] > 2.326)  power_COMPOSITE_DAC3++;
    if  (COMPOSITE_DAC6[j] > 2.326)  power_COMPOSITE_DAC6++;

    if (IBD1_EDAC[j] > 2.326 ) power_IBD1_EDAC++;
    if (IBD2_EDAC[j] > 2.326 ) power_IBD2_EDAC++;


    /* for the maximized composite test*/
    /* empirical critical values for each mixed chi-sqaured variable*/
    if  (composite_DS4[j] > 7.251)  power_composite_DS4++;
    if  (COMPOSITE_DAC4[j] > 8.681)  power_COMPOSITE_DAC4++;
    if  (COMPOSITE_DAC5[j] > 9.976)  power_COMPOSITE_DAC5++;


}

  power_VC /=N_samples;
  power_original_HE /=N_samples;
  power_sum_HE /=N_samples;
```

```
power_new_HE /=N_samples;
power_Xu_HE /=N_samples;
power_Viss_HE /=N_samples;
power_Forrest_HE /=N_samples;
power_Sham_HE /=N_samples;
power_robust_Sham /=N_samples;
power_score_standard_pos /=N_samples;
power_score_robust_pos /=N_samples;
power_score_3 /=N_samples;
power_score_4 /=N_samples;
power_score_5 /=N_samples;
power_score_6 /=N_samples;
power_meanIBD1_pos /=N_samples;
power_meanIBD1_neg /=N_samples;
power_meanIBD2_pos /=N_samples;
power_meanIBD2_neg /=N_samples;
power_meanIBD3_pos /=N_samples;
power_meanIBD3_neg /=N_samples;
power_composite1 /=N_samples;
power_composite2 /=N_samples;
power_composite3 /=N_samples;
power_composite_CS2 /=N_samples;
power_composite_CS3 /=N_samples;
power_COMPOSITE_DAC11 /=N_samples;
power_COMPOSITE_DAC12 /=N_samples;
power_COMPOSITE_DAC2 /=N_samples;
power_COMPOSITE_DAC3 /=N_samples;
power_COMPOSITE_DAC6 /=N_samples;
power_IBD1_EDAC /=N_samples;
power_IBD2_EDAC /=N_samples;
power_composite_DS4 /=N_samples;
power_COMPOSITE_DAC4 /=N_samples;
power_COMPOSITE_DAC5 /=N_samples;

/*for average value of each test statistic*/
for ( j = 0; j < N_samples; j++) {

  ave_VC += LL_store[j] ;
  ave_original_HE += original_HE[j] ;

  ave_sum_HE += sum_HE[j];
  ave_new_HE += new_HE[j];
  ave_Xu_HE += Xu_HE[j];
  ave_Viss_HE += Viss_HE[j];
  ave_Forrest_HE += Forrest_HE[j] ;
  ave_Sham_HE += Sham_HE[j] ;
  ave_robust_Sham += robust_Sham[j] ;
  ave_score_standard += score_standard[j] ;
  ave_score_robust += score_robust[j] ;
  ave_score_3 += score_3[j] ;
  ave_score_4 += score_4[j] ;
  ave_score_5 += score_5[j] ;
  ave_score_6 += score_6[j] ;

  ave_meanIBD1 += meanIBD1[j];
  ave_meanIBD2 += meanIBD2[j];
  ave_composite1 += composite1[j];
```

```
  ave_composite2 += composite2[j];

  ave_meanIBD3 += meanIBD3[j];
  ave_composite3 += composite3[j];
  ave_composite_CS2 += composite_CS2[j];
  ave_composite_CS3 += composite_CS3[j];
  ave_COMPOSITE_DAC11 += COMPOSITE_DAC11[j];
  ave_COMPOSITE_DAC12 += COMPOSITE_DAC12[j];
  ave_COMPOSITE_DAC2 += COMPOSITE_DAC2[j];
  ave_COMPOSITE_DAC3 += COMPOSITE_DAC3[j];
  ave_COMPOSITE_DAC6 += COMPOSITE_DAC6[j];
  ave_IBD1_EDAC += IBD1_EDAC[j];
  ave_IBD2_EDAC += IBD2_EDAC[j];

  ave_composite_DS4 += composite_DS4[j];
  ave_COMPOSITE_DAC4 += COMPOSITE_DAC4[j];
  ave_COMPOSITE_DAC5 += COMPOSITE_DAC5[j];
}

ave_VC /= N_samples ;
ave_original_HE /= N_samples ;
ave_sum_HE /= N_samples ;
ave_new_HE /= N_samples ;
ave_Xu_HE /= N_samples ;
ave_Viss_HE /= N_samples ;
ave_Forrest_HE /= N_samples ;
ave_Sham_HE /= N_samples ;
ave_robust_Sham /= N_samples ;
ave_score_standard /= N_samples ;
ave_score_robust /= N_samples ;
ave_score_3 /= N_samples ;
ave_score_4 /= N_samples ;
ave_score_5 /= N_samples ;
ave_score_6 /= N_samples ;

ave_meanIBD1 /= N_samples ;
ave_meanIBD2 /= N_samples ;
ave_meanIBD3 /= N_samples ;
ave_composite1 /= N_samples ;
ave_composite2 /= N_samples ;
ave_composite3 /= N_samples ;
ave_composite_CS2 /= N_samples ;
ave_composite_CS3 /= N_samples ;
ave_COMPOSITE_DAC11 /= N_samples ;
ave_COMPOSITE_DAC12 /= N_samples ;
ave_COMPOSITE_DAC2 /= N_samples ;
ave_COMPOSITE_DAC3 /= N_samples ;
ave_COMPOSITE_DAC6 /= N_samples ;
ave_IBD1_EDAC /= N_samples ;
ave_IBD2_EDAC /= N_samples ;

ave_composite_DS4 /= N_samples ;
ave_COMPOSITE_DAC4 /= N_samples ;
ave_COMPOSITE_DAC5 /= N_samples ;


double var_VC = 0.0 ;
```

```
  double var_original_HE = 0.0 ;
  double var_sum_HE = 0.0 ;
  double var_new_HE = 0.0 ;
  double var_Xu_HE = 0.0 ;
  double var_Viss_HE = 0.0 ;
  double var_Forrest_HE = 0.0 ;
  double var_Sham_HE = 0.0 ;
  double var_robust_Sham = 0.0 ;
  double var_score_standard = 0.0 ;
  double var_score_robust = 0.0 ;
  double var_score_3 = 0.0 ;
  double var_score_4 = 0.0 ;
  double var_score_5 = 0.0 ;
  double var_score_6 = 0.0 ;

  double var_meanIBD1 = 0.0 ;
  double var_meanIBD2 = 0.0 ;
  double var_composite1 = 0.0 ;
  double var_composite2 = 0.0 ;

  double var_meanIBD3 = 0.0 ;
  double var_composite3 = 0.0 ;
  double var_composite_CS2 = 0.0 ;
  double var_composite_CS3 = 0.0 ;
  double var_COMPOSITE_DAC11 = 0.0 ;
  double var_COMPOSITE_DAC12 = 0.0 ;
  double var_COMPOSITE_DAC2 = 0.0 ;
  double var_COMPOSITE_DAC3 = 0.0 ;
  double var_COMPOSITE_DAC6 = 0.0 ;
  double var_IBD1_EDAC = 0.0;
  double var_IBD2_EDAC = 0.0;

  double var_composite_DS4 = 0.0 ;
  double var_COMPOSITE_DAC4 = 0.0 ;
  double var_COMPOSITE_DAC5 = 0.0 ;

  for (j=0; j <N_samples; j++){
    /*loop over all studies to calculate empirical variances */
   var_VC += (LL_store[j]-ave_VC)* (LL_store[j]-ave_VC)/(N_samples-1);
   var_original_HE += (original_HE[j] -ave_original_HE)* (original_HE[j] -
ave_original_HE)/(N_samples-1);
   var_sum_HE += (sum_HE[j]-ave_sum_HE)* (sum_HE[j]-ave_sum_HE)/(N_samples-
1);
   var_new_HE += (new_HE[j]-ave_new_HE)* (new_HE[j]-ave_new_HE)/(N_samples-
1);
   var_Viss_HE += (Viss_HE[j]-ave_Viss_HE)* (Viss_HE[j]-
ave_Viss_HE)/(N_samples-1);
   var_Xu_HE += (Xu_HE[j]-ave_Xu_HE)* (Xu_HE[j]-ave_Xu_HE)/(N_samples-1);
   var_Forrest_HE += (Forrest_HE[j]-ave_Forrest_HE)* (Forrest_HE[j]-
ave_Forrest_HE)/(N_samples-1);
   var_Sham_HE += (Sham_HE[j]-ave_Sham_HE)* (Sham_HE[j]-
ave_Sham_HE)/(N_samples-1);
   var_robust_Sham += (robust_Sham[j]-ave_robust_Sham)* (robust_Sham[j]-
ave_robust_Sham)/(N_samples-1);
   var_score_standard += (score_standard[j]-ave_score_standard)*
(score_standard[j]-ave_score_standard)/(N_samples-1);
```

```
   var_score_robust += (score_robust[j]-ave_score_robust)* (score_robust[j]-
ave_score_robust)/(N_samples-1);
   var_score_3 += (score_3[j]-ave_score_3)* (score_3[j]-
ave_score_3)/(N_samples-1);
   var_score_4 += (score_4[j]-ave_score_4)* (score_4[j]-
ave_score_4)/(N_samples-1);
   var_score_5 += (score_5[j]-ave_score_5)* (score_5[j]-
ave_score_5)/(N_samples-1);
   var_score_6 += (score_6[j]-ave_score_6)* (score_6[j]-
ave_score_6)/(N_samples-1);

   var_meanIBD1 += (meanIBD1[j]- ave_meanIBD1)* (meanIBD1[j]-
ave_meanIBD1)/(N_samples-1);
   var_meanIBD2 += (meanIBD2[j]- ave_meanIBD2)* (meanIBD2[j]-
ave_meanIBD2)/(N_samples-1);
   var_composite1 += (composite1[j]- ave_composite1)* (composite1[j]-
ave_composite1)/(N_samples-1);
   var_composite2 += (composite2[j]- ave_composite2)* (composite2[j]-
ave_composite2)/(N_samples-1);
   var_meanIBD3 += (meanIBD3[j]- ave_meanIBD3)* (meanIBD3[j]-
ave_meanIBD3)/(N_samples-1);
   var_composite3 += (composite3[j]- ave_composite3)* (composite3[j]-
ave_composite3)/(N_samples-1);
   var_composite_CS2 += (composite_CS2[j]- ave_composite_CS2)*
(composite_CS2[j]- ave_composite_CS2)/(N_samples-1);
   var_composite_CS3 += (composite_CS3[j]- ave_composite_CS3)*
(composite_CS3[j]- ave_composite_CS3)/(N_samples-1);

var_COMPOSITE_DAC11 += (COMPOSITE_DAC11[j]- ave_COMPOSITE_DAC11)*
(COMPOSITE_DAC11[j]- ave_COMPOSITE_DAC11)/(N_samples-1);
var_COMPOSITE_DAC12 += (COMPOSITE_DAC12[j]- ave_COMPOSITE_DAC12)*
(COMPOSITE_DAC12[j]- ave_COMPOSITE_DAC12)/(N_samples-1);

var_COMPOSITE_DAC2 += (COMPOSITE_DAC2[j]- ave_COMPOSITE_DAC2)*
(COMPOSITE_DAC2[j]- ave_COMPOSITE_DAC2)/(N_samples-1);
var_COMPOSITE_DAC3 += (COMPOSITE_DAC3[j]- ave_COMPOSITE_DAC3)*
(COMPOSITE_DAC3[j]- ave_COMPOSITE_DAC3)/(N_samples-1);
var_COMPOSITE_DAC6 += (COMPOSITE_DAC6[j]- ave_COMPOSITE_DAC6)*
(COMPOSITE_DAC6[j]- ave_COMPOSITE_DAC6)/(N_samples-1);

var_IBD1_EDAC += (IBD1_EDAC[j]-ave_IBD1_EDAC)* (IBD1_EDAC[j]-
ave_IBD1_EDAC)/(N_samples-1);
var_IBD2_EDAC += (IBD2_EDAC[j]-ave_IBD2_EDAC)* (IBD2_EDAC[j]-
ave_IBD2_EDAC)/(N_samples-1);

var_COMPOSITE_DAC4 += (COMPOSITE_DAC4[j]- ave_COMPOSITE_DAC4)*
(COMPOSITE_DAC4[j]- ave_COMPOSITE_DAC4)/(N_samples-1);
var_COMPOSITE_DAC5 += (COMPOSITE_DAC5[j]- ave_COMPOSITE_DAC5)*
(COMPOSITE_DAC5[j]- ave_COMPOSITE_DAC5)/(N_samples-1);

var_composite_DS4 += (composite_DS4[j]- ave_composite_DS4)*
(composite_DS4[j]- ave_composite_DS4)/(N_samples-1);

} //end of loop for variance calcu.
```

```
  ////////////////////////////////////////
  /*output into file "powers"              */
  ////////////////////////////////////////
  fprintf(fp5,"\nStatistic                  Power    Mean
Standard_deviation\n");
  fprintf(fp5,"Variance_Components       %lf %lf %lf \n",
power_VC,ave_VC,sqrt(var_VC));
  fprintf(fp5,"ORIGINAL.HE               %lf %lf %lf \n",
        power_original_HE, ave_original_HE,sqrt(var_original_HE));
  fprintf(fp5,"TRAIT.SUM                 %lf %lf %lf \n",
power_sum_HE,ave_sum_HE,sqrt(var_sum_HE));
  fprintf(fp5,"TRAIT.PRODUCT             %lf %lf %lf\n",
power_new_HE,ave_new_HE,sqrt(var_new_HE));
  fprintf(fp5,"V&H                       %lf %lf %lf\n",
power_Viss_HE,ave_Viss_HE,sqrt( var_Viss_HE));
  fprintf(fp5,"FORREST                   %lf %lf %lf\n",
power_Forrest_HE,ave_Forrest_HE,sqrt(var_Forrest_HE));
  fprintf(fp5,"XU                        %lf %lf %lf \n",
power_Xu_HE,ave_Xu_HE,sqrt(var_Xu_HE));
  fprintf(fp5,"HE.COM-correlation        %lf %lf %lf\n",
power_Sham_HE,ave_Sham_HE,sqrt(var_Sham_HE));
  fprintf(fp5,"HE.COM-combination        %lf %lf %lf\n",
        power_robust_Sham, ave_robust_Sham,sqrt(var_robust_Sham));
  fprintf(fp5,"SCORE1                    %lf %lf %lf \n",
power_score_standard_pos, ave_score_standard, sqrt(var_score_standard));
  fprintf(fp5,"SCORE2                    %lf %lf %lf \n",
power_score_robust_pos, ave_score_robust,sqrt(var_score_robust));
  fprintf(fp5,"SCORE3                    %lf %lf %lf \n",
        power_score_3, ave_score_3,sqrt(var_score_3));
  fprintf(fp5,"SCORE4                    %lf %lf %lf\n",
        power_score_4, ave_score_4,sqrt(var_score_4));
  fprintf(fp5,"SCORE5                    %lf %lf %lf \n",
          power_score_5, ave_score_5,sqrt(var_score_5));
  fprintf(fp5,"SCORE6(RDP)               %lf %lf %lf \n",
          power_score_6, ave_score_6,sqrt(var_score_6));


  fprintf(fp5,"IBD1_CS(pos)              %lf %lf %lf\n",
        power_meanIBD1_pos, ave_meanIBD1,sqrt(var_meanIBD1));
  fprintf(fp5,"IBD2_CS                   %lf %lf %lf\n",
          power_meanIBD2_pos, ave_meanIBD2,sqrt(var_meanIBD2));
  fprintf(fp5,"IBD3_CS                   %lf %lf %lf\n",
        power_meanIBD3_pos, ave_meanIBD3,sqrt(var_meanIBD3));
  fprintf(fp5,"IBD1_DS(neg)              %lf %lf %lf\n",
          power_meanIBD1_neg, ave_meanIBD1,sqrt(var_meanIBD1));
  fprintf(fp5,"IBD2_DS                   %lf %lf %lf\n",
          power_meanIBD2_neg, ave_meanIBD2,sqrt(var_meanIBD2));
  fprintf(fp5,"IBD3_DS                   %lf %lf %lf\n",
        power_meanIBD3_neg, ave_meanIBD3,sqrt(var_meanIBD3));

  fprintf(fp5,"COMPOSITE_DS1             %lf %lf %lf\n",
          power_composite1, ave_composite1,sqrt(var_composite1));
  fprintf(fp5,"COMPOSITE_DS2             %lf %lf %lf\n",
          power_composite2, ave_composite2,sqrt(var_composite2));
  fprintf(fp5,"COMPOSITE_DS3             %lf %lf %lf\n",
          power_composite3, ave_composite3,sqrt(var_composite3));
```

```
  fprintf(fp5,"COMPOSITE_DS4(Maximized)  %lf %lf %lf\n",
          power_composite_DS4, ave_composite_DS4,sqrt(var_composite_DS4 ));
  fprintf(fp5,"COMPOSITE_CS2             %lf %lf %lf\n",
          power_composite_CS2, ave_composite_CS2,sqrt(var_composite_CS2));
  fprintf(fp5,"COMPOSITE_CS3             %lf %lf %lf\n",
          power_composite_CS3, ave_composite_CS3,sqrt(var_composite_CS3));


  fprintf(fp5,"IBD1_EDAC(Gu)             %lf %lf %lf\n",
power_IBD1_EDAC, ave_IBD1_EDAC,sqrt(var_IBD1_EDAC));
  fprintf(fp5,"IBD2_EDAC                 %lf %lf %lf\n",
power_IBD2_EDAC, ave_IBD2_EDAC,sqrt(var_IBD2_EDAC));
  fprintf(fp5,"3part-composite           %lf %lf %lf\n",
power_COMPOSITE_DAC3, ave_COMPOSITE_DAC3,sqrt(var_COMPOSITE_DAC3));
  fprintf(fp5,"RDP-composite             %lf %lf %lf\n",
power_COMPOSITE_DAC6, ave_COMPOSITE_DAC6,sqrt(var_COMPOSITE_DAC6));

  fprintf(fp5,"COMPOSITE_DAC11           %lf %lf %lf\n",
power_COMPOSITE_DAC11, ave_COMPOSITE_DAC11,sqrt(var_COMPOSITE_DAC11));
  fprintf(fp5,"COMPOSITE_DAC12           %lf %lf %lf\n",
power_COMPOSITE_DAC12, ave_COMPOSITE_DAC12,sqrt(var_COMPOSITE_DAC12));
  fprintf(fp5,"COMPOSITE_DAC2            %lf %lf %lf\n",
power_COMPOSITE_DAC2, ave_COMPOSITE_DAC2,sqrt(var_COMPOSITE_DAC2));
  fprintf(fp5,"COMPOSITE_DAC4(Max-3part) %lf %lf %lf\n",
power_COMPOSITE_DAC4, ave_COMPOSITE_DAC4,sqrt(var_COMPOSITE_DAC4));
  fprintf(fp5,"COMPOSITE_DAC5(Max-4part) %lf %lf %lf\n",
power_COMPOSITE_DAC5, ave_COMPOSITE_DAC5,sqrt(var_COMPOSITE_DAC5));


  fprintf(fp5,"\nThe number of families in each study: %d\n",num_fam);
  fprintf(fp5,"The number of studies: %d\n",N_samples);

  fclose(fp5);}
```

APPENDIX E

EXAMPLE R PROGRAM

```
##############################################################################
# Weight_DAC_3.R
# This program read in file “user_COMPOSITE” and find optimal weights for
# combining score6_DS and IBD_CS to form the “RDP composite”

DAC.stat<-read.table("user_COMPOSITE")
attach(DAC.stat)

#V1 to V6 are: HE_DS,IBD2_DS,S&P.HE_CS,IBD2_CS,original.HE_CS2,score6_DS
# V1,V2,V5,V6 should be negative; V3,V4 should be positive.
# Exploring weights by F&F2000 method (they used N=10,000 ; n=200)
# This program finds optimal weights for combining DS_component(V6) with
# CS_component(V4) to get highest power (and mean values) of the final
composite # statistic – tau


nt<-90
theta<-seq(0,pi/2,len=nt)
theta2<-theta*180/pi

# Third loop
component1<- (-1)*V6
component2<- V4

comp.EDAC<-NULL
for (i in 1:nt)
{
    comp.EDAC<-cbind( comp.EDAC,
component1*cos(theta[i])+component2*sin(theta[i]))
}


power.fun<-function(y)
{
    sum(y>=2.326)/(length(y))
}

power<-apply(comp.EDAC,2,power.fun)
```

```
ave.EDAC<-apply(comp.EDAC,2,mean)

opt.by.power<-cbind(theta,theta2,power,ave.EDAC)
max.power<-power==max(power)
opt.by.power[max.power,]
tau.by.power <- opt.by.power[max.power,]
tau <- opt.by.power[max.power,1]
angles<-cbind(tau,tau*180/pi)

w1<-cos(tau)
w2<-sin(tau)
weights<-cbind(w1,w2)

print(angles)
print(weights)
print(tau.by.power)


output.list<-list(angles,weights,tau.by.power)
print(output.list)
names(DAC.stat)<-c("ORIGINAL.HE.DS","IBD.DS","SP1.CS","IBD.CS",
"ORIGINAL.HE.CS","SCORE6.DS")
print(summary(DAC.stat))


sink("Test.Wgt3",append=F)
print(output.list)
cat("summary of component statistics\n")
print(summary(DAC.stat))
sink()

# ./a.out; R --slave < Weight_DAC_3.R
# Splus --slave --nosave < filename
```

APPENDIX F

GLOSSARY

allele - One of the alternative versions of a gene at a given location (locus) along a chromosome.

ascertainment - The pedigree recruitment through an affected sibling, proband.

complex trait - A trait whose mode of inheritance does not follow the Medelian laws.

autosomal - Any of the chromosomes other than the sex-determining chromosomes (i.e., the X and Y) or the genes on these chromosomes.

correlation-based statistic - Any statistic that draws power from the correlation between IBD-sharing and trait value similarity in a sample of pedigrees.

combination statistic - Any statistic that draws power from both the correlation and the IBD-sharing information.

complex trait - A trait whose mode of inheritance does not follow the Medelian laws.

composite statistic - Combination statistic proposed by Forrest and Feingold (2000) that is a weighted sum of a correlation-based statistic and an IBD-sharing statistic.

dominant - A disease is transmitted in a dominant fashion when only one copy of the disease allele is required to cause disease.

EDSP - Extreme discordant sibling pairs.

EDAC - Extreme discordant and concordant sibling pairs. Our use of this term encompasses any sampling scheme that uses both discordant and concordant pairs, regardless of the extremity.

frequency - The rate at which a particular allele is expected to be found in the general population.

gamete - Sperm or egg cell.

gene - The basic unit of heredity, consisting of a segment of DNA arranged in a linear manner along a chromosome, which codes for a specific protein or segment of protein leading to a particular characteristic or function.

genetic marker - An identifiable segment of DNA (e.g., RFLP, VNTR, microsatellite) with enough variation between individuals that its inheritance and co-inheritance with alleles of a given gene can be traced; used in linkage analysis.

genotype - The specific set of alleles inherited at a locus

genotyping - Testing that reveals the specific alleles inherited by an individual; particularly useful for situations in which more than one genotypic combination can produce the same clinical presentation, as in the ABO blood group, where both the AO and AA genotypes yield type A blood.

heterozygote - An individual with two different alleles.

homozygote - An individual with two identical alleles.

identical by descent (IBD) - Alleles in an individual or in two people are identical because they have been transmitted from the same common ancestor.

identical by state (IBS) - Coincidental possession of identical alleles in an individual or in two people.

IBD-sharing statistic - Any statistic that draws power from the marginal IBD-sharing information.

linkage - The tendency for genes or segments of DNA closely positioned along a chromosome to segregate together at meiosis and therefore be inherited together.

linkage methods - Methods for mapping genes that use family data and try to detect co-segregation of traits and shared genetic material within families

locus - The physical site or location of a specific gene on a chromosome.

meiosis - The process of generating gametes.

MDSP - Moderately discordant sibling pairs.

multiple-proband sampling - Samples collected on the basis of two or more people in each pedigree having particular phenotypes.

optimal sampling - A strategy that precisely ascertains the most informative pairs in the population, proposed by Purcell et al (2001).

pedigree - A family.

phenotype - The observable physical and/or biochemical characteristics of the expression of a gene; the clinical presentation of an individual with a particular genotype.

population sample - Samples collected in a way that is not dependent on the trait values.

proband - The individual through whom a family is ascertained.

quantitative trait - Any human trait that is measured on a numerical scale.

quantitative trait locus (QTL) - Locus that influences quantitative traits.

recessive - a disease is transmitted in a recessive fashion when two copies of the disease allele are required to cause disease.

recombination - The exchange of a segment of DNA between two homologous chromosomes during meiosis leading to a novel combination of genetic material in the offspring.

recombination fraction - The probability of occurrence of a recombinant event (usually denoted as θ).

score statistic – Statistic based on the derivative of the variance components likelihood. Can be either correlation-based or combination.

selected sample – Samples collected based on the trait values of one or more members, including both single-proband and multiple-proband sampling.

single-proband sampling - Samples collected on the basis of one member (proband) in each pedigree having particular phenotypes.

BIBLIOGRAPHY

Abecasis GR, Cherny SS, Cookson WO and Cardon LR (2002) Merlin-rapid analysis of dense genetic maps using sparse gene flow trees. Nat Genet 30:97-101

Abney M, Ober C, McPeek MS (2002) Quantitative-trait homozygosity and association mapping and empirical genomewide significance in large, complex pedigrees: fasting serum-insulin level in the Hutterites. Am J Hum Genet 70:920-934

Alarcon M, Cantor RM, Liu J, Gilliam TC, Geschwind DH (2002) Autism Genetic Research Exchange Consortium. Evidence for a language quantitative trait locus on chromosome 7q in multiplex autism families. Am J Hum Genet 70:60-71

Alcais A, Abel L (2000) Linkage analysis of quantitative trait loci: sib pairs or sibships? Hum Hered 50:251-256

Allison DB, Fernanadez JR, Heo M, Beasley TM (2000) Testing the robustness of the new Haseman-Elston quantitative-trait loci-mapping procedure. Am J Hum Gen 67:249-252

Allison DB, Heo M, Schork NJ, Wong SL, Elston RC (1998) Extreme selection strategies in gene mapping studies of oligogenic quantitative traits do not always increase power. Hum Hered 48(2):97-107

Allison DB, Neale MC, Zannolli R, Schork NJ, Amos CI, Blangero J (1999) Testing the robustness of the likelihood-ratio test in a variance-component quantitative-trait loci-mapping procedure. Am J Hum Genet 65:531-544

Almasy L, Blangero J (1998) Multipoint quantitative-trait linkage analysis in general pedigrees. Am J Hum Genet 62:1198-1211

Almasy L, Blangero J (1998) Multipoint quantitative-trait linkage analysis in general pedigrees. Am J Hum Genet 62:1198-1211

Almasy L, Dyer TD, Blangero J (1997) Bivariate quantitative trait linkage analysis: pleiotropy versus co-incident linkages. Genet Epidemiol 14:953-958

Amos CI (1994) Robust variance-components approach for assessing genetic linkage in pedigrees. Am J Hum Genet 54:535-543

Arya R, Duggirala R, Jenkinson CP, Almasy L, Blangero J, O'Connell P, Stern MP (2004) Evidence of a Novel Quantitative-Trait Locus for Obesity on Chromosome 4p in Mexican Americans. Am J Hum Genet 74:272-282

Atwood LD, Heard-Costa NL, Cupples LA, Jaquish CE, Wilson PW, D'Agostino RB (2002) Genomewide linkage analysis of body mass index across 28 years of the Framingham Heart Study. Am J Hum Genet 71:1044-1050

Blackwelder WC, Elston RC (1985) A comparison of sib-pair linkage tests for disease susceptibility loci. Genet Epidemiol 2:85-97

Cai G, Li T, Deng H, Zhao J, Hu X, Murray RM, Liu X, Sham PC, Collier DA (2001) Affected sibling pair linkage analysis of qualitative and quantitative traits for schizophrenia on chromosome 22 in a Chinese population. Am J Med Genet 105:321-327

Cardon LR, Fulker DW (1994) The power of interval mapping of quantitative trait loci, using selected sib pairs. Am J Hum Genet 55:825-833

Carey G and Williamson J (1991) Linkage analysis of quantitative traits: increased power by using selected samples. Am J Hum Genet 49:786-796

Chen W-M, Broman KW, Liang K-Y (2004) Quantitative trait linkage analysis by generalized estimating equations: Unification of variance components and Haseman-Elston regression. Genet Epidemiol 26(4): 265-272

Chen W-M, Broman KW, Liang K-Y (*Submitted*) Power and robustness of linkage tests for quantitative traits in general pedigrees. Available at www.biostat.jhsph.edu/~broman

Davis S, Weeks DE (1997) Comparison of nonparametric statistics for detection of linkage in nuclear families: single marker evaluation. Am J Hum Genet 61:1431–1444

Deng H-W, Deng H, Liu Y-J, Liu Y-Z, Xu F-H, Shen H, Conway T, Li J-L, Huang, QY, Davies KM, Recker RR (2002) A Genomewide Linkage Scan for Quantitative-Trait Loci for Obesity Phenotypes. Am J Hum Genet 70:1138-1151

DeStefano AL, Lew MF, Golbe LI, Mark MH, Lazzarini AM, Guttman M, Montgomery E et al. (2002) PARK3 influences age at onset in Parkinson disease: a genome scan in the GenePD study. Am J Hum Genet 70:1089-1095

Dolan CV, Boomsma DI (1998) Optimal selection of sib pairs from random samples for linkage analysis of a QTL using the EDAC test. Behav Genet 28(3):197-206
Dong C, Wang S, Li WD, Li D, Zhao H, Price RA (2003) Interacting genetic loci on chromosomes 20 and 10 influence extreme human obesity. Am J Hum Genet 72:115-124

Drigalenko E (1998) How sib pairs reveal linkage. Am J Hum Genet 63:1242-1245

Eaves L, Meyer J (1994) Locating human quantitative trait loci: guidelines for the selection of sibling pairs for genotyping. Behav Genet 24:443-455

Econs MJ, Koller DL, Hui SL, Fishburn T, Conneally PM, Johnston CC Jr., Peacock M, Foroud TM (2004) Confirmation of Linkage to Chromosome 1q for Peak Vertebral Bone Mineral Density in Premenopausal White Women. Am J Hum Genet 74:223-228

Elston RC, Buxbaum S, Jacobs KB, Olson JM (2000) Haseman and Elston Revisited. Genet Epidemiol 19:1-17

Feingold E (2001) Methods for linkage analysis of quantitative trait loci in humans.
Theor Pop Biol 60:167-180

Feingold E (2002) Regression-based QTL mapping in the 21st century. Am J Hum Genet 71:217-222

Feitosa MF, Borecki IB, Rich SS, Arnett DK, Sholinsky P, Myers RH, Leppert M, Province MA (2002) Quantitative-trait loci influencing body-mass index reside on chromosomes 7 and 13: the National Heart, Lung, and Blood Institute Family Heart Study. Am J Hum Genet 70:72-82

Fisher SE, Francks C, McCracken JT, McGough JJ, Marlow AJ, MacPhie IL, Newbury DF, Crawford LR, Palmer CG, Woodward JA, Del'Homme M, Cantwell DP, Nelson SF, Monaco AP, Smalley SL (2002) A genomewide scan for loci involved in attention-deficit/hyperactivity disorder. Am J Hum Genet 70:1183-1196

Forrest W and Feingold E (2000) Composite statistics for QTL mapping with moderately discordant sibling pairs. Am J Hum Genet 66:1642-1660

Forrest W (2001) Weighting improves the new Haseman-Elston method. Hum Hered 52:47-54

Fox CS, Cupples LA, Chazaro I, Polak JF, Wolf PA, D'Agostino RB, Ordovas JM, O'Donnell CJ (2004) Genomewide Linkage Analysis for Internal Carotid Artery Intimal Medial Thickness: Evidence for Linkage to Chromosome 12. Am J Hum Genet 74:253-261

Francks C, Fisher SE, MacPhie IL, Richardson AJ, Marlow AJ, Stein JF, Monaco AP (2002) A genomewide linkage screen for relative hand skill in sibling pairs. Am J Hum Genet 70:800-805

Fullerton J, Cubin M, Tiwari H, Wang C, Bomhra A, Davidson S, Miller S, Fairburn C, Boodwin G, Neale MC, Fiddy S, Mott R, Allison DB, Flint J (2003) Linkage analysis of extremely discordant and concordant sibling pairs identifies quantitative-trait loci that influence variation in the human personality trait neuroticism. Am J Hum Genet 72:879-890

Garner CP, Tatu T, Best S, Creary L, Thein SL (2002) Evidence of genetic interaction between the beta-globin complex and chromosome 8q in the expression of fetal hemoglobin. Am J Hum Genet 70:793-799

Ghosh S, Reich T (2002) Integrating sibship data for mapping quantitative trait loci. Ann Hum Genet 66:169-182

Goldstein DR, Dudoit S, Speed TP (2001) Power and robustness of a score test for linkage analysis of quantitative traits using identity by descent data on sib pairs. Gen Epi 20:415-431

Gu C, Rao DC (1997a) A linkage strategy for detection of human quantitative-trait loci. I. Generalized relative risk ratios and power of sib pairs with extreme trait values. Am J Hum Genet 61(1):200-10.

Gu C, Rao DC (1997b) A linkage strategy for detection of human quantitative-trait loci. II. Optimization of study designs based on extreme sib pairs and generalized relative risk ratios. Am J Hum Genet 61(1):211-22.

Gu C, Todorov AA, Rao DC (1996) Combining extremely concordant sibpairs with extremely discordant sibpairs provides a cost effective way to linkage analysis of QTLs. Genet Epidemiol 13:513-533

Gudbjartsson DF, Jonasson K, Kong CA (1999) Fast Multipoint Linkage Calculation with Allegro. Am J Hum Genet suppl 65

Hall MA, Norman PJ, Thiel B, Tiwari H, Peiffer A, Vaughan RW, Prescott S, Leppert M, Schork NJ, Lanchbury JS (2002) Quantitative trait loci on chromosomes 1, 2, 3, 4, 8, 9, 11, 12, and 18 control variation in levels of T and B lymphocyte subpopulations. Am J Hum Genet 70:1172-1182

Haseman JK, Elston RC (1972) The investigation of linkage between a quantitative trait and a marker locus. Behav Genet 2:3-19

Huang J, Jiang Y (2003) Genetic linkage analysis of a dichotomous trait incorporating a tightly linked quantitative trait in affected sibling pairs. Am J Hum Genet 72:949-960

Kaplan DE, Gayan J, Ahn J, Won TW, Pauls D, Olson RK, DeFries JC, Wood F, Pennington BF, Page GP, Smith SD, Gruen JR (2002) Evidence for linkage and association with reading disability on 6p21.3-22. Am J Hum Genet 70:1287-1298

Knapp M (1998) Evaluation of a restricted likelihood ratio test for mapping quantitative trait loci with extreme discordant sib pairs. Ann Hum Genet 62:75-87

Kruglyak L, Daly MJ, Reeve-Daly MP, Lander ES (1996) Parametric and nonparametric linkage analysis: a unified multipoint approach.Am J Hum Genet 58(6):1347-1363.

Kruse R, Seuchter SA, Baur MP, Knapp M (1997) The “possible triangle” test for extreme discordant sib pairs. Genet Epidemiol 14:833-838

Li C, Scott LJ, Boehnke M (2004) Assessing Whether an Allele Can Account in Part for a Linkage Signal: The Genotype-IBD Sharing Test (GIST). Am J Hum Genet 74:418-431

Li Z, Gastwirth JL (2001) A weighted test using both extreme discordant and concordant sib pairs for detecting linkage. Genet Epidemiol 20(1):34-43.

Li Z, Zhang H (2000) Mapping quantitative trait loci in humans using both extreme discordant and concordant sib pairs: a unified approach for meta-analysis. Commun Stat Theory Methods 29: 1115-27

Lin JL, Hayden MR, Almqvist EW, Brinkman RR, Durr A, Dode C, Morrison PJ et al. (2003) A genome scan for modifiers of age at onset in Huntington disease: The HD MAPS study. Am J Hum Genet 73:682-687

Morton NE (1959) Genetics tests under incomplete ascertainment. Am J Hum Genet 11:116

O'Brien EK, Zhang X, Nishimura C, Tomblin JB, Murray JC (2003) Association of specific language impairment (SLI) to the region of 7q31. Am J Hum Genet 2:1536-1543

Pajukanta P, Allayee H, Krass KL, Kuraishy A, Soro A, Lilja HE, Mar R, Taskinen MR, Nuotio I, Laakso M, Rotter JI, de Bruin TW, Cantor RM, Lusis AJ, Peltonen L (2003) Combined analysis of genome scans of dutch and finnish families reveals a susceptibility locus for high-density lipoprotein cholesterol on chromosome 16q. Am J Hum Genet 72:903-917

Palmer LJ, Buxbaum SG, Larkin E, Patel SR, Elston RC, Tishler PV, Redline S. A (2003) whole-genome scan for obstructive sleep apnea and obesity. Am J Hum Genet 72:340-350

Palmer LJ, Jacobs KB, Elston RC (2000) Haseman and Elston revisited: The effects of ascertainment and residual familial correlations on the power to detect linkage. Gen Epi 19:456-460

Purcell S, Cherny SS, Hweitt JK, Sham PC (2001) Optimal sibship selection for genotyping in quantitative trait locus linkage analysis. Hum Hered 52:1-13

Putter H, Sandkuijl LA, van Houwelingen JC (2002) Score test for detecting linkage to quantitative traits. Genet Epidemiol 22:345-335

Risch N, Zhang H (1995) Extreme discordant sib pairs for mapping quantitative trait loci in humans. Science 268:1584-1589

Risch N, Zhang H (1996) Mapping quantitative trait loci with extreme discordant sib pairs: Sampling considerations. Am J Hum Genet 58(4):836-843
Rogus JJ, Harrington DP, Jorgenson E, Xu X (1997) Effectiveness of extreme discordant sib pairs to detect oligogenic disease loci. Genet Epidemiol 14:879-884

S.A.G.E. (2002) Statistical Analysis for Genetic Epidemiology. Computer program package available from Statistical Solutions Ltd, Cork, Ireland.

Sengul H, Weeks DE, Feingold E (2001) A survey of affected-sibship statistics for nonparametric linkage analysis. Am J Hum Genet 69: 179-190

Sham, P. C., Zhao, J. H., Cherny, S. S. and Hewitt, J. K. (2000) Variance components QTL linkage analysis of selected and non-normal samples: conditioning on trait values. Genet Epidemiol 19:S22–S28.

Sham PC, Purcell S (2001) Equivalence between Haseman-Elston and variance components linkage analyses for sib pairs. Am J Hum Genet 68:1527-1532

Sham PC, Purcell S, Cherny SS, Abecasis GR (2002) Powerful regression-based quantitative–trait linkage analysis of general pedigrees. Am J Hum Genet 71:238-253

Shete S, Jacobs K, Elston R (2003) Adding further power to the Haseman and Elston method for detecting linkage in larger sibships: weighting sums and differences. Hum Hered 55:79-85

Silverman EK, Palmer LJ, Mosley JD, Barth M, Senter JM, Brown A, Drazen JM, Kwiatkowski DJ, Chapman HA, Campbell EJ, Province MA, Rao DC, Reilly JJ, Ginns LC, Speizer FE, Weiss ST (2002) Genomewide linkage analysis of quantitative spirometric phenotypes in severe early-onset chronic obstructive pulmonary disease. Am J Hum Genet 70:1229-1239

Slager SL, Schaid DJ, Cunningham JM, McDonnell SK, Marks AF, Peterson BJ,

Hebbring SJ, Anderson S, French AJ, Thibodeau SN (2003) Confirmation of linkage of prostate cancer aggressiveness with chromosome 19q. Am J Hum Genet 72:759-762

Soria JM, Almasy L, Souto JC, Bacq D, Buil A, Faure A, Martinez-Marchan E, Mateo J, Borrell M, Stone W, Lathrop M, Fontcuberta J, Blangero J (2002) A quantitative-trait locus in the human factor XII gene influences both plasma factor XII levels and susceptibility to thrombotic disease. Am J Hum Genet 70:567-574

Stein CM, Schick JH, Taylor HG, Shriberg LD, Millard C, Kundtz-Kluge A, Russo K, Minich N, Hansen A, Freebairn LA, Elston RC, Lewis BA, Iyengar SK (2004) Pleiotropic Effects of a Chromosome 3 Locus on Speech-Sound Disorder and Readin. Am J Hum Genet 74:283-297

Szatkiewicz JP, T Cuenco K, Feingold E (2003) Recent advances in human quantitative-trait-locus mapping: comparison of methods for discordant sibling pairs. Am J Hum Genet 73(4):874-85

Szatkiewicz JP and Feingold E (*submitted*) A new linkage statistic for discordant sibling pairs outperforms current statistics.

Szatkiewicz JP and Feingold E (*submitted*) QTL mapping with discordant and concordant sibling pairs - new statistics and new design strategies.

Tang H-K, Siegmund D (2001) Mapping quantitative trait loci in oligogenic models. Biostat 2:147-162

Todorov AA, Province MA, Borecki IB, Rao DC (1997) Trade-off between sibship size and sampling scheme for detecting quantitative trait loci. Hum Hered 47:1-5

T Cuenco K, Szatkiewicz JP, Feingold E (2003) Recent advances in human quantitative-trait-locus mapping: comparison of methods for selected sibling pairs. Am J Hum Genet 73(4):863-73

van Asselt KM, Kok HS, Putter H, Wijmenga C, Peeters PHM, van der Schouw YT, Grobbee DE, te Velde ER, Mosselman S, Pearson PL (2004) Linkage Analysis of Extremely Discordant and Concordant Sibling Pairs Identifies Quantitative Trait Loci Influencing Variation in Human Menopausal Age. Am J Hum Genet 74:444-453

Visscher PM, Hopper JL (2001) Power of regression and maximum likelihood methods to map QTL from sib-pair and DZ twin data. Ann Hum Genet 65:583-601

Wang K (2002) Efficient score statistics for mapping quantitative trait loci with extended pedigrees. Hum Hered 54:57-68

Wang K, Huang J (2002a) A score-statistic approach for the mapping of quantitative-trait loci with sibships of arbitrary size. Am J Hum Genet 70:412-424

Wang K, Huang J (2002b) Score test for mapping quantitative-trait loci with sibships of arbitrary size when the dominance effect is not negligible. Genet Epidemiol 23:398-412

Watanabe et al. (2000). The Finland United States Investigation of NonInsulin-Dependent Diabetes Mellitus Genetics (FUSION) Study. II. An Autosomal Genome Scan for Diabetes-Related Quantitative-Trait Loci. Am J Hum Genet 67:1186-1200.

Williams JT et al. (1999a) Joint multipoint analysis of multivariate qualitative and quantitative traits I. Am J Hum Genet 65:1134-1147

Williams JT et al. (1999b) Joint multipoint analysis of multivariate qualitative and quantitative traits II. Am J Hum Genet 65:1148-1160

Wilson SG, Reed PW, Bansal A, Chiano M, Lindersson M, Langdown M, Prince RL, Thompson D, Thompson E, Bailey M, Kleyn PW, Sambrook P, Shi MM, Spector TD (2003) Comparison of Genome Screens for Two Independent Cohorts Provides Replication of Suggestive Linkage of Bone Mineral Density to 3p21 and 1p36. Am J Hum Genet 72:144-155

Wiltshire S, Frayling TM, Hattersley AT, Hitman GA, Walker M, Levy JC, O'Rahilly S, Groves CJ, Menzel S, Cardon LR, McCarthy MI (2002) Evidence for linkage of stature to chromosome 3p26 in a large U.K. Family data set ascertained for type 2 diabetes. Am J Hum Genet 543-546

Whittemore AS, Halpern J (1994) A class of tests for linkage using affected pedigree members. Biometrics 50:118-127

Wu X, Cooper RS, Borecki I, Hanis C, Bray M, Lewis CE, Zhu X, Kan D, Luke A, Curb D (2002) A combined analysis of genomewide linkage scans for body mass index from the

National Heart, Lung, and Blood Institute Family Blood Pressure Program. Am J Hum Genet 70:1247-1256

Xu J, Bleecker ER, Jongepier H, Howard TD, Koppelman GH, Postma DS, Meyers DA (2002) Major recessive gene(s) with considerable residual polygenic effect regulating adult height: confirmation of genomewide scan results for chromosomes 6, 9, and 12. Am J Hum Genet 71:646-650

Xu X, Rogus JJ, Terwedow HA, Yang J, Wang Z, Chen C, Niu T, et al. (1999) An extreme-sib-pair genome scan for genes regulating blood pressure. Am J Hum Genet 64:1694-1701

Xu X, Weiss S, Xu X, Wei LJ (2000) A unified Haseman-Elston method for testing linkage with quantitative traits. Am J Hum Genet 67:1025-1028

Yu X, Knott SA, Visscher PM (2004) Theoretical and empirical power of regression and maximum-likelihood methods to map quantitative trait loci in general pedigrees. Am J Hum Genet 75(1):17-26

Zhang W, Collins A, Lonjou C, Morton NE (2002) A linkage tournament: affection status, parametric analysis, multivariate traits, and enhancements to variance components and relative pairs. Ann Hum Genet 66:87-98

Zhang H, Leckman JF, Pauls DL, Tsai C-P, Kidd KK, Campos MR and the Tourette Syndrome Association International Consortium for Genetics (2002) Genomewide scan of hoarding in sib pairs in which both sibs have Gilles de la Tourette syndrome. Am J Hum Genet 70:896-904.

Zhang W, Tapper W, Collins A, Jacobs KB, Elston RC (2001) A tournament of linkage tests in complex inheritance. Hum Hered 52:140-148