2009

Comparison of Exact Methods for Analyzing Family-Based Samples

Abbie Lundgreen
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/gradreports

Part of the Statistics and Probability Commons

Recommended Citation
Lundgreen, Abbie, "Comparison of Exact Methods for Analyzing Family-Based Samples" (2009). All Graduate Plan B and other Reports. 1233.
https://digitalcommons.usu.edu/gradreports/1233

This Report is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Plan B and other Reports by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.
Comparison of Exact Methods for Analyzing Family-Based Samples

by

Abbie Lundgreen

A report submitted in partial fulfillment
of the requirements for the degree
of
MASTER OF SCIENCE
in
Statistics

Approved:

Dr. Chris Corcoran
Major Professor

Dr. John Stevens
Committee Member

Dr. Maria Norton
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2009
Family-based association tests are used to identify genes that increase the risk of developing a disease, while controlling for spurious associations caused by population structure. The exact family-based association test, exact FBAT, is a unified approach which can be applied to tests of different genetic models, sampling designs, null hypotheses, and missing parental information.

The purpose of this report is to compare the power of the exact FBAT with two other tests, exact conditional logistic regression (CLR) and the exact trend test for clustered data (QEM). Pedigrees of sibships were simulated based upon a variety of different parameters, and then the test statistic was calculated for each. Examining the power for each test, we find that QEM is clearly the most powerful test among the three in detecting linkage among data from sampled sibships. The difference in power among exact FBAT and exact CLR is small, with exact CLR demonstrating a slight advantage over exact FBAT. While the relative differences in power is substantial for small sample sizes, the gaps shrink as the number of families increases.
CONTENTS

ABSTRACT..................................................................................................................... ii

CONTENTS.................................................................................................................. iii

INTRODUCTION............................................................................................................. 1

METHODS....................................................................................................................... 7

  2.1 Exact FBAT............................................................................................................. 7
  2.2 Exact Conditional Logistic Regression ................................................................. 9
  2.3 Exact Trend Test for Clustered Data..................................................................... 10
  2.4 Programming......................................................................................................... 11

RESULTS......................................................................................................................... 13

CONCLUSION.................................................................................................................. 17

REFERENCES.................................................................................................................. 18

APPENDIX....................................................................................................................... 21

  Appendix I. R Code for Data Simulation................................................................. 21
  Appendix II. SAS Script for CLR............................................................................. 26
  Appendix III. C Code for QEM................................................................................. 27
  Appendix IV. Perl Code to Run Entire Simulation................................................. 30
1 Introduction

Epidemiology is the study of disease patterns in a population and the factors that are associated with causation of disease, with a primary goal of disease prevention. Genetic epidemiology focuses more specifically on the role of inherited causes of disease within families and populations. In addition to detecting heritability patterns of a particular disease, genetic epidemiology also aims to localize the gene and find a marker associated with disease susceptibility.

Conventionally, the search for risk-inducing genes has been preceded by analyses that establish the degree to which a given disease appears to be inherited, along with the pattern of heritability. Such studies require family-based samples (of complex pedigrees, nuclear families, twins, or adopted children), in order to assess the aggregation of disease within families, and to distinguish between potential environmental and genetic components underlying the disease. Evidence of heritability can then motivate a search for specific genes or other genetic markers that either lie on the causal pathway to disease (i.e., the gene or marker affects the regulation of a biological function that modifies disease risk), or are at least physically proximate to a disease-related marker.

The completion of the Human Genome and Human HapMap Projects has dramatically altered epidemiologic approaches to identifying genetically determined disease risks. For example, more comprehensive understanding of the human genome has increased the availability of microsatellite DNA polymorphisms, such as single base-pair mutations along the DNA strand referred to as single nucleotide polymorphisms (or SNPs). This has somewhat recently led to a marked increase in so-called hypothesis-free searches, using panels of thousands (now commonly hundreds of thousands) of SNPs across the genomes of those participating in a given study. The objective of such an investigation is to identify all marker variants associated with the phenotype or disease of interest.

On the surface, assessing genetic association with complex diseases is comparable to any traditional data analysis: the manifestation of a genetic marker can be viewed as simply a fixed covariate used to explain variation in any physical trait (called a phenotype) measured on a study subject. However, there are some analytic problems unique to genetic association that have been recently addressed by a significant body of statistical work. These developments have occurred somewhat in tandem over the past few decades with
progress in laboratory technology.

To identify a marker that is associated with disease risk, an effective approach is to examine whether the gene is linked to a known marker. *Linkage* refers to the physical proximity of an investigative marker to a gene whose function affects the risk for a given phenotype. In a traditional epidemiologic setting, using population-based or case-control samples, we can examine whether the marker appears in affected individuals more often than would be expected if the marker and the disease were independent; in other words, to identify markers that are associated with the disease. However, such associations are not always the result of linkage. In fact, genetic associations arise from three main causes. First, the marker itself may be contributing to disease susceptibility. Second, the marker may be in *linkage disequilibrium* with the disease gene. Third, the association may be attributable to population structure. In the latter case finding a genetic association does not assist in establishing the proximity of a marker and a disease gene.

Linkage disequilibrium is a consequence of the molecular process of cell division and reduction called meiosis, which yields reproductive cells that combine during fertilization to initiate the development of a new organism. Linkage disequilibrium occurs when a disease-causing gene is physically proximate to an investigative marker. If the marker of a diseased individual undergoes a molecular change, then that resulting mutation will more likely be passed to the individual's offspring along with the disease-causing allele. Because of the phenomenon of *recombination*, where paired chromosomes swap sections of DNA somewhat randomly during meiosis, a marker and its linked disease-causing genetic allele will not with certainty be passed together from a parent to a child. However, a physically close tandem will be passed together with greater probability. These biological processes have significant implications for data analysis. Borecki and Suarez (2001) discuss the statistical roles of linkage and recombination, and their use in analyzing family based studies. Though linkage disequilibrium does not necessarily imply linkage, in the literature linkage disequilibrium often connotes the presence of both linkage and association. Linkage disequilibrium in this sense is found only over relatively small physical distances over a chromosome, since recombination occurs more frequently over larger distances, causing linkage disequilibrium between two widely separated genes to decay rapidly. Therefore, when there is evidence of disequilibrium between an investigative marker and a disease,
it strongly suggests that the disease gene is close to the marker.

In contrast to linkage disequilibrium, an association that results from population structure gives no information about the relative genetic positions of a marker and the disease locus. Population structure is the result of a population comprised of genetically distinct subgroups. In any subpopulation there may exist allelic variants and phenotypic traits that are overrepresented relative to the general population. Thus a gene and trait may be associated despite the lack of any biological relationship. (An illustration of potentially spurious associations with the human leukocyte antigen complex—the HLA gene—within an ethnically heterogeneous population is discussed by Lander and Schork, 1996.) While observational studies are often successful in establishing associations, population structure can render their results difficult to reproduce.

These circumstances yield certain trade-offs with regard to study design: case-control or case-cohort designs often require relatively less cost, but are susceptible to problems of population structure. Family-based studies—using samples of sibships, parent-child combinations, or complex pedigrees—may be logistically more difficult, but effectively eliminate potential confounding due to population structure. Several studies have moreover compared population- and family-based designs with respect to required sample size and statistical power, although conclusions vary depending on assumptions made about the availability of parental data when families are sampled.

Since the location of genes can be established by demonstrating linkage with known markers, statistical methods for linkage analysis have received much attention over the past several decades. These methods have been broadly classified as parametric versus nonparametric. In either case they rely on data sampled from families, effectively eliminating confounding due to population structure. The most widely used approach to parametric linkage analysis is the method of lod (short for “log-odds”) scores, which is likelihood-based and thus requires several assumptions about the underlying genetic inheritance model. This technique is implemented by collecting pedigree data for the gene and marker of interest, then calculating lod scores as the ratio of the probability of realizing the observed pedigree to the probability of observing a pedigree assuming no linkage. In common with other techniques of linkage analysis, lod scores require pedigrees having multiple siblings. Unfortunately, for complex diseases (such as diabetes or obesity) whose
etiology involves a relatively large number of genetic and environmental factors, parametric linkage methods suffer from low power (for example, see Elston, 2002, and Ott, 2001). Hence, a drawback of these methods is that they can only detect linkage over relatively large regions of the genome, making it difficult to implicate specific genes in relation to disease.

In contrast to parametric methods such as lod scores, the collective approach of family based association tests yields greater power in finding linkage, while requiring fewer assumptions about the underlying genetic model. The distinguishing feature of family-based designs is the use of parental genotypes and simple Mendelian inheritance rules to characterize the distribution of alleles that are passed from parents to children with disease. Building on methods introduced by Ott (1989), Terwilliger and Ott (1992), Rubinstein et al. (1981), and Falk and Rubinstein (1987), Spielman and Ewens (1993) introduced the simple but subsequently widely used and studied transmission disequilibrium test (TDT). The TDT requires sampled trios of two genotyped parents and one child with disease, and compares the transmission rates of two marker alleles. Under the null hypothesis of no linkage and no association, the two investigative alleles for a given marker should be transmitted in equal proportion. The introduction of the TDT led to a number of extensions, as the design constraint of sampling parent/child trios can be somewhat impractical, particularly for outcomes related to advanced age such as Alzheimer's disease. These variations on the TDT were mainly derived to allow for alternative designs involving missing parents, multiple alleles, and unaffected children. A thorough description of the TDT and its extensions is given by Ewens and Spielman (1995).

The significant body of methodologic work spawned by the TDT demonstrates the utility of family-based methods. However, the TDT and its extensions are each constrained by a sampling strategy involving some specific combination of parent/child data. The unified framework for family based tests (FBAT) developed by Rabinowitz and Laird (2000) builds on the idea of the TDT to allow for different family structures, arbitrary disease phenotypes, and the presence or absence of parental genotypes. To use FBAT, one need only define the genetic marker and the trait of interest, so that FBAT effectively subsumes nearly all TDT-type tests as special cases. FBAT software is freely available through a website located at http://www.biosat.harvard.edu/~fbat/default.html.
Under currently available implementations, inferences using FBAT are based mainly on large-sample distributional assumptions regarding the FBAT statistic. However, as the FBAT approach is conditional, these assumptions are used to approximate the exact distribution, which is fully specified under the null hypothesis. Current versions of FBAT do not include the exact test, although using the web-based software an investigator can obtain a Monte Carlo approximation to the exact test under certain settings.

Exact tests are generally recommended for analyses involving small or sparse data sets. Such data certainly arise in family-based settings, when few families are sampled, or when rare investigative phenotypes or genetic alleles render the testing distribution more discrete. However, there are several even more compelling reasons – unique to the family-based setting – for computing the actual permutation distribution. For example, due to the multiplicity of tests when assessing linkage with many genetic markers, investigators often employ very small significance levels. For some applications, Blacker et al. (1998) suggest a nominal significance level on the order of $10^{-6}$. This represents a commonly used Bonferroni-type correction that controls the overall false positive rate by dividing 0.05 by the number of investigative markers. Looking at such extreme critical regions necessitates even greater accuracy in p-value computation. Moreover, so-called genome-wide screening, where thousands (and in the near future perhaps millions) of genetic markers are scanned to find evidence that some are proximate to disease-causing genes, suggests a more omnibus permutation approach. Notably, Van Steen et al. (2005) point out that this application has especially critical implications as population-based studies are able to include this kind of broad screening for an increased number of markers at a decreased cost. Overarching all of these issues, there is building evidence that exact inference in the FBAT setting is often more powerful than the corresponding asymptotic approximations. This power advantage becomes significantly more pronounced as more alleles are considered.

The large investment required by genomewide association studies makes it imperative that selected methods for subsequent statistical analyses are as powerful as possible. A common criticism of the FBAT approach is that the required conditioning makes it more conservative, relative to statistical power for detecting linkage and association. If a family-based sample consists of nuclear families or sibships, then there are potential
alternatives to FBAT that require less conditioning. For example, the sibship transmission disequilibrium test (or S-TDT) proposed by Spielman and Ewens (1998) for studies of siblings with no parental data is a generalized Mantel-Haenszel test (see Laird et al., 1998). That is, it can be formulated as a stratified Cochran-Armitage trend test, which can be evaluated within the framework of a conditional logistic regression (CLR) where each stratum (i.e., family) has a unique intercept. In addition, Corcoran et al. (2002) suggest a trend test for clustered data, based on a quadratic exponential model (QEM) that includes a term to model the overdispersion arising from within-family correlations. Exact tests based on these three options (FBAT, CLR, and QEM) require conditioning that would appear to rank from greatest to least, in terms of the information sacrificed to obtain the associated permutation distribution. For this project, we wish to conduct a simulation study in order to compare the relative power of these approaches.
2 Methods

In this section, we describe the three exact methodological approaches that we wish to compare with respect to their power for detecting linkage among data from sampled sibships. First, we summarize how to obtain the permutation distribution for the FBAT statistic of Laird and Rabinowitz (2000). We then consider two variations of exact conditional logistic regression that account for within-family clustering, using stratification in one case (CLR) and a single overdispersion parameter in the other (QEM). Finally, we outline how we carried out the programming of the simulation study in Section 2.4 in order to evaluate the competing methods.

2.1 Exact FBAT

As mentioned in Section 1, the general family-based association test (FBAT) approach effectively subsumes the TDT and its extensions as special cases. By choosing an appropriate formulation of the test statistic, FBAT can handle pedigree data under a variety of conditions, including arbitrary patterns of missing information, multiple alleles, and arbitrary phenotypes.

The test statistic proposed by Rabinowitz and Laird (2000) has the form

\[ S = \sum_{i=1}^{N} \sum_{j=1}^{K} X_{ij} T_{ij}, \]

where \( i \) indexes the families, \( j \) indexes the offspring within each family, \( X_{ij} \) denotes the genotype of the \((i,j)\)th child, and \( T_{ij} \) represents a function of the phenotype for the \((i,j)\)th child. In this section we assume that there are two possible manifestations (alleles) of the investigative gene denoted generically by A and B. If A is the allele of interest, then under an additive model we would code \( X_{ij} \) as the number of A alleles carried by the \((i,j)\)th subject, so that \( X_{ij} \in \{0,1,2\} \). A dominant model yields \( X_{ij} = 1 \) if the \((i,j)\)th subject has at least one A allele, and a recessive model results in \( X_{ij} = 1 \) only if the \((i,j)\)th subject carries two copies of the A allele (i.e., the \((i,j)\)th subject is homozygous for A).

To illustrate, we will assume an additive model, and a dichotomous trait such that \( T_{ij} = 1 \) if the \((i,j)\)th subject is affected and \( T_{ij} = 0 \) otherwise. Under these conditions, \( S \) represents the number of A alleles transmitted to the affected offspring. A larger value
of \( S \) provides greater evidence of linkage and association, as more A alleles are observed to be passed to affected offspring than unaffected offspring. A key to computing the distribution of \( S \) is conditioning on minimal sufficient statistics in order to eliminate nuisance parameters. Rabinowitz and Laird (2000) point out that the conditioning must include all phenotypes along with the sufficient statistics for parental genotypes – either the observed parental genotypes themselves, or a function of the observed genotypes of all subjects if parental data are missing. Subject to these constraints, we can compute the conditional expectation and variance of \( S \) under the null hypothesis of no linkage and no association. Under appropriate conditions, the standardized test statistic follows an approximate standard normal distribution under the null.

While the large-sample normal approximation is computationally convenient and used widely in practice, for reasons discussed in the previous section it is often desirable to compute the exact tail area. In fact, specific cases of exact family based tests are not without precedent. For example, the TDT procedure of Spielman et al. (1993) is a version of McNemar's test, which can be solved exactly using the binomial distribution. The so-called sibling-transmission disequilibrium test (S-TDT) – proposed by Spielman and Ewens (1998) for studies of siblings with no parental data – is a generalized Mantel-Haenszel test (see Laird et al., 1998), and its exact distribution can be obtained by using permutation procedures for stratified \( 2 \times K \) tables. However, these exact methods can be applied only to specific study designs.

Although the FBAT implementation is largely based on the large-sample distribution of the statistic (1), the permutation distribution is obtainable. To this end, Rabinowitz and Laird (2000) outline the required conditioning in detail, describing the minimal sufficient statistics under general family based designs and giving the distribution of the offspring genotypes for any family configuration. To illustrate, consider a family with both parental genotypes unknown and three children whose genotypes are AA, AB, and BB. The conditional distribution of the offspring genotypes is given by randomly assigning the genotypes AA, AB and BB independently among the children, with probabilities 1/4, 1/2, and 1/4, respectively, discarding any outcome in which either AA or BB is not assigned.

To understand the exact approach, consider first a simple case involving a single family
Table 1: Tabular representation of the genotypes of a sampled family with \( K \) siblings.

<table>
<thead>
<tr>
<th>Allele</th>
<th>Child (( j ))</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>( x_1 ) 2 ( x_2 ) \cdots ( x_K )</td>
<td>( m )</td>
</tr>
<tr>
<td>B</td>
<td>( 2-x_1 ) ( 2-x_2 ) \cdots ( 2-x_K )</td>
<td>( 2K-m )</td>
</tr>
<tr>
<td>Total</td>
<td>( 2 ) ( 2 ) \cdots ( 2 )</td>
<td>( 2K )</td>
</tr>
</tbody>
</table>

With \( K \) children and a marker having two alleles, A and B. The distribution of the alleles among the offspring in such a family can be conveniently represented in tabular form as shown in Table 1. For the present, we focus on a single family, dropping the subscript \( i \). Hence, under the additive model, for example, \( x_j \) represents the number of A alleles possessed by the \( j \)th child. The value \( t_j \) represents the phenotype “score” for the \( j \)th child (e.g., for a dichotomous trait \( t_j = 1 \) if the \( j \)th child is affected and \( t_j = 0 \) otherwise).

This tabulation is a critical step in our conceptualization, allowing us to link the exact FBAT problem to more conventional permutation methods for categorical data. Under the additive model the column margins are always 2, since each child carries only two alleles of the investigative marker. The first row margin \( m \) is equal to the total number of A alleles observed across all of the siblings.

A permutation test is carried out by first finding every table that can be constructed subject to the constraints imposed by the conditioning. We denote this conditional reference set by \( \Gamma \). Each table in \( \Gamma \) has an associated value of the test statistic (1), along with a probability under the null as computed according to the tables given in Rabinowitz and Laird (2000). An exact p-value is computed by summing together the probabilities of all tables in \( \Gamma \) that have a value of the test statistic at least as great as the observed data.

2.2 Exact Conditional Logistic Regression

As indicated in Section 1, a version of the TDT suggested for sibships by Spielman and Ewens (1998) for studies of siblings with no parental data is a generalized Mantel-
Haenszel test (see Laird et al., 1998), which can be formulated within the framework of a conditional logistic regression (CLR) where each stratum (i.e., family) has a unique intercept. Given \( N \) families, let \( Y_{ij} = 1 \) if the \( j \)th individual in the \( i \)th family is affected, and \( Y_{ij} = 0 \) otherwise. For \( x_{ij} \), a \( p \)-dimensional vector of covariates, define:

\[
\pi_{ij} = Pr(Y_{ij} = 1|x_{ij}).
\]

Then the logistic regression model is of the form:

\[
\log \left( \frac{\pi_{ij}}{1 - \pi_{ij}} \right) = \gamma_i + x_{ij} \beta,
\]

where \( \gamma_i \) is a scalar parameter specific to the stratum (i.e., representing the baseline disease rate within the \( i \)th family), \( x_{ij} \) is a function of the genotype for the \( j \)th individual in the \( i \)th family, and \( \beta \) is the corresponding genotypic effect.

Since the focus of inference is \( \beta \), either conditional likelihood-based or exact inference can be accomplished by conditioning on the sufficient statistics for the \( \gamma_i \), which are regarded as nuisance parameters. Conditional asymptotic distribution of the Wald, likelihood ratio, or score statistics can subsequently be used for testing \( H_0 : \beta = 0 \). The permutation distributions for these statistics can be obtained by conditioning on the sufficient statistics corresponding to the \( N \) nuisance intercepts represented by the \( \gamma_i \). For each family, this statistic is given by the number of individuals with disease.

### 2.3 Exact Trend Test for Clustered Data

The exact trend test for clustered data proposed by Corcoran et al. (2000) (referred to here as QEM, since its basis is a quadratic exponential model) appears to be another useful alternative to both FBAT and CLR. Note that for stratified logistic regression there are \( N \) sufficient statistics that require conditioning, corresponding to the \( N \) stratum-specific nuisance intercepts that need to be eliminated to obtain the permutation distribution. For the QEM, we need only condition on two sufficient statistics: one for the common intercept and another for an overdispersion parameter that accounts for the within-cluster correlation.

To summarize the formulation for this approach, consider the \( i \)th family, in which there are \( k_i \) subjects and \( x_{ij} \) represents a function of the genotype for \( j \)th family member.
Letting \( Y_{ij} \) represent the binary disease status for the \( j \)th sibling, then \( Z_i = \sum k_{ij} Y_{ij} \) represents the number of siblings with disease in the \( i \)th family. The density of \( Y_{ij} \) can be expressed as

\[
Pr(Y_i = y_i) = \exp\{\theta_i z_i - \delta_i z_i (k_i - z_i) + A_i(\theta_i, \delta_i)\},
\]

where \( \delta_i \) is the dispersion parameter representing the correlation within the \( i \)th family, and \( A_i(\theta_i, \delta_i) \) is the normalizing constant. Assuming exchangeability of the responses across all families we have \( \delta_i = \delta \). Using a logistic link such that \( \theta_i = \alpha + \beta x_{ij} \), and further assuming independent clusters, the joint density across all families can therefore be expressed as

\[
Pr(Z = z|x) = \prod_{i=1}^{N} \left( \begin{array}{c} k_i \\ z_i \end{array} \right) \exp \left\{ \alpha s_1 + \beta t - \delta s_2 + \sum_{i=1}^{N} A_i(\alpha, \beta, \delta) \right\},
\]

where \( s_1 = \sum z_i, t = \sum x_i z_i \) and \( s_2 = \sum z_i (k_i - z_i) \). An attractive feature of this model is that it reduces to standard logistic regression where \( \delta = 0 \) (i.e., with no clustering).

Because the density is of the exponential family, \( s_1, t, \) and \( s_2 \) are sufficient for \( \alpha, \beta, \) and \( \delta \). By conditioning on both \( s_1 \) and \( s_2 \), we can eliminate all unknown parameters and obtain the exact distribution of \( Z \) under the null. Given the number of clusters and size of each, we can find all possible tables, which give rise to the permutation distribution of the sufficient statistic \( t \) for \( \beta \). The exact p-value can be obtained from the tails of this distribution.

### 2.4 Programming

In order to carry out the analysis, data composed of family-based samples was simulated using the R software package. The source code is included in this paper as Appendix I. Exact FBAT was implemented using software that is freely available through a website located at http://www.math.usu.edu/~schneit/efbat/. SAS®1 software was utilized in order to implement CLR and the source code is attached as Appendix II. The third analytic approach, QEM, was implemented via a code written in the C language. The entire analysis was then run for a large number of iterations in batch mode in order

---

1SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. © indicates USA registration.
to find the approximate power of each test using the perl programming language and a significance level of 0.05. Both source codes are attached as Appendix III and IV.
3 Results

To compare the operating characteristics of the methods outlined in Section 2, we carried out a simulation study under various commonly observed conditions. Family-based samples were generated using the R software package, which we then analyzed using each approach as implemented in Section 2.4. We consider two alleles $A$ and $B$ at an investigative marker locus, where $A$ is the risk-modifying variant. For all analyses, we assumed an additive effect, such that disease risk increases with number of $A$ alleles.

Generating data from families requires assumptions about several parameter values, including (i) overall disease prevalence $K$ within the sampling population; (ii) the allele frequency $PA$, specified so that the sum of each allele frequency must be equal to one; and (iii) the attributable fraction $AF$, representing the reduction of disease prevalence in the absence of the risk-inducing allele. $K$ was allowed to assume values of 0.01 and 0.05, $PA$ assumed values of 0.2 and 0.5, and $AF$ remained constant a 0.5. We carried out the study using varying numbers of sampled families (50, 100, 500), but in every case we assume three children (siblings) with no parental data. Pedigree files were created by R and output into text files, formatted individually for the programs used for each type of analysis.

From Table 2, it’s apparent that QEM is the most powerful test for detecting linkage among the sibships, regardless of the frequency of the allele of interest or the prevalence of disease in the population. Exact CLR appears to be only slightly more powerful than exact FBAT.

In general, the frequency of the allele of interest has an inverse effect with the power of each test. More explicitly, as the frequency in which the allele of interest is present in the pedigree increases, the power of the test decreases, and vice versa. On the other hand, as the number of families per data set increases, the power also increases. The relative differences in power between the three exact tests also shrinks as the sample size increases. Thus, the advantage of one test over another decreases as the sample size increases, especially for pedigrees in which the disease prevalence is higher ($K=0.05$) and allele frequency is low ($PA=0.1$).
<table>
<thead>
<tr>
<th>Num.Fam</th>
<th>PA</th>
<th>K</th>
<th>exact FBAT</th>
<th>exact CLR</th>
<th>QEM</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.1</td>
<td>0.01</td>
<td>0.0010</td>
<td>0.0014</td>
<td>0.0189</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.0899</td>
<td>0.0934</td>
<td>0.2222</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.01</td>
<td>0.0006</td>
<td>0.0004</td>
<td>0.0117</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.0617</td>
<td>0.0630</td>
<td>0.1046</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td>0.0006</td>
<td>0.0010</td>
<td>0.0033</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.0277</td>
<td>0.0278</td>
<td>0.0485</td>
</tr>
<tr>
<td>100</td>
<td></td>
<td>0.1</td>
<td>0.0083</td>
<td>0.0068</td>
<td>0.0629</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.3096</td>
<td>0.3098</td>
<td>0.4488</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.01</td>
<td>0.0047</td>
<td>0.0050</td>
<td>0.0355</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.1839</td>
<td>0.1868</td>
<td>0.2609</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td>0.0040</td>
<td>0.0036</td>
<td>0.0150</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.0667</td>
<td>0.0662</td>
<td>0.1127</td>
</tr>
<tr>
<td>500</td>
<td></td>
<td>0.1</td>
<td>0.2890</td>
<td>0.3042</td>
<td>0.4464</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.9839</td>
<td>0.9852</td>
<td>0.9855</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.01</td>
<td>0.1718</td>
<td>0.1848</td>
<td>0.2504</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.8611</td>
<td>0.8690</td>
<td>0.8921</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td>0.0613</td>
<td>0.0667</td>
<td>0.1035</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.3302</td>
<td>0.3420</td>
<td>0.5134</td>
</tr>
</tbody>
</table>

Table 2: Powers for exact FBAT, exact CLR, and QEM.
<table>
<thead>
<tr>
<th>Num.Fam</th>
<th>PA</th>
<th>K</th>
<th>FBAT</th>
<th>Logistic</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>0.1</td>
<td>0.01</td>
<td>0.0151</td>
<td>0.0192</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.2006</td>
<td>0.2032</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.01</td>
<td>0.0139</td>
<td>0.0160</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.1375</td>
<td>0.1366</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td>0.0105</td>
<td>0.0138</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.0696</td>
<td>0.0700</td>
</tr>
<tr>
<td>100</td>
<td>0.1</td>
<td>0.01</td>
<td>0.0491</td>
<td>0.0472</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.4239</td>
<td>0.4196</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.01</td>
<td>0.0422</td>
<td>0.0438</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.2665</td>
<td>0.2682</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td>0.0294</td>
<td>0.0320</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.1079</td>
<td>0.1064</td>
</tr>
<tr>
<td>500</td>
<td>0.1</td>
<td>0.01</td>
<td>0.3944</td>
<td>0.3944</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.9892</td>
<td>0.9892</td>
</tr>
<tr>
<td></td>
<td>0.2</td>
<td>0.01</td>
<td>0.2542</td>
<td>0.2542</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.8899</td>
<td>0.8899</td>
</tr>
<tr>
<td></td>
<td>0.5</td>
<td>0.01</td>
<td>0.0996</td>
<td>0.0996</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.05</td>
<td>0.3765</td>
<td>0.3765</td>
</tr>
</tbody>
</table>

Table 3: Powers for FBAT and CLR.
According to Table 3, the power comparison for FBAT and CLR is very similar to the power comparison among the exact tests. CLR is more powerful in detecting linkage among sibships than FBAT for the majority of the simulations, although the difference is negligible.
4 Conclusions

The use of genome-wide association studies to find genetic variants associated with the development of particular diseases has increased with the availability of microsatellite DNA polymorphisms. While these tests may be applied to either population or family-based studies, the latter is often preferred as they avoid the issue of confounding due to population structure. Given the logistical demands of family-based studies, sample sizes are often small or sparse, preventing the usual model assumptions from being met. Consequently, exact tests are generally recommended over their asymptotic counterparts. Within these family-based studies, thousands of genetic markers are analyzed simultaneously for genetic linkage, which introduces the issue of multiple testing. A Bonferroni-type correction is often applied to control the Type-I error rate, making it more difficult to find significant results. Thus, it is important to apply the most powerful exact test in order to maximize the probability of finding novel associations.

In comparing exact FBAT, exact CLR, and QEM in terms of their power to detect linkage among data from sampled sibships, we found that QEM is clearly the most powerful test and should be considered as an alternative to exact FBAT. The performances of the other tests are similar, with exact CLR demonstrating a slight advantage. Thus, the loss of information due to conditioning does, indeed, appear to play an important role in our power calculations.

Although this simulation was conducted without the use of any parental information, it may be interesting to compare the ability of the three tests to detect linkage among data collected from sibships in which one or both parental genotypes are also known.
REFERENCES


Http://www.math.usu.edu/~schneit/efbat/.


Appendix I. R Code for Data Simulation

```r
pedisimu=function(familynum, childnum, PA, K, AF, model, verbose=FALSE){
  # generate parents' genotype
  child=rep(0, childnum)
  c=childnum*familynum
  childID=rep(0,c)
  familyID=rep(0,c)
  fatherID=rep(0,c)
  motherID=rep(0,c)
  affectstat=rep(0,c)
  sex=rep(0,c)
  allel.1=rep(0,c)
  allel.2=rep(0,c)
  marker=rep(0,c)
  PB=1-PA
  j=1
  countO=0
  count1=0
  count2=0
  diseaseO=0
  disease1=0
  disease2=0
  while(j<=familynum){
    parent=rep(0,2)
    parent.allel=runif(2)
    for(i in 1:2){
      if(parent.allel[i]<((PA*PA)){parent[i]=2}
      if(parent.allel[i]<((1-PB*PB)&&parent.allel[i]>=(PA*PA)){parent[i]=1}
      if(parent.allel[i]>=(1-PB*PB)){parent[i]=0}
    }
    # determine offsprings' genotype
      for(i in 1:childnum){child[i]=2}
    }
    if(parent[1]==0&&parent[2]==0){
      for(i in 1:childnum){child[i]=2}
    }
  }
}
```
for(i in 1:childnum){child[i]=0}
}
for(i in 1:childnum){child[i]=1}
}
for(i in 1:childnum){
    child.allel=runif(1)
    if(child.allel<0.5){
        child[i]=2
    }else{child[i]=1}
}
}
for(i in 1:childnum){
    child.allel=runif(1)
    if(child.allel<0.25){
        child[i]=2
    }if((child.allel>=0.25)&&(child.allel<0.75)){
        child[i]=1
    }else{child[i]=0}
}
}
for(i in 1:childnum){
    child.allel=runif(1)
    if(child.allel<0.5){
        child[i]=0
    }else{child[i]=1}
}
}
#calculate penetrance function
qO=K*(1-AF)
#additive model
if (model=="add"){
q2=qO+(K-q0)/PA
q1=(q0+q2)/2
}
#dominant model

```r
if(model=="dom"){
  q2=((1-PA)^2*q0-K)/((1-PA)^2-1)
  q1=q2
}
```

#recessive model

```r
if(model=="rec"){
  q2=(K-q0)/PA+q0
  q1=q0
}
```

```r
sumaffect=0
```

#determine the affection status

```r
for(i in 1:childnum){
  k=childnum*(j-1)+i
  familyID[k]=j
  fatherID[k]=100*j+0
  motherID[k]=100*j+1
  childID[k]=k
  #determine the sex
  s=runif(1)
  if(s<0.5){sex[k]=1}
  else{sex[k]=2}
  if(child[i]==0){
    affect=runif(1)
    if(affect<q0){affectstat[k]=1}
    else{affectstat[k]=0}
    allel.1[k]=2
    allel.2[k]=2
  }
  if(child[i]==1){
    affect=runif(1)
    if(affect<q1){affectstat[k]=1}
    else{affectstat[k]=0}
    allel.1[k]=1
    allel.2[k]=2
  }
  if(child[i]==2){
    affect=runif(1)
```

```r
23
```
if(affect<q2){affectstat[k]=1}
else{affectstat[k]=0}
allel.1[k]=1
allel.2[k]=1

if(allel.1[k]==2&&allel.2[k]==2){
    marker[k]=0
    count0=count0+1
    if(affectstat[k]==1){
        disease0=disease0+1
    }
}

if((allel.1[k]==1&&allel.2[k]==2)||(allel.1[k]==2&&allel.2[k]==1)){
    marker[k]=1
    count1=count1+1
    if(affectstat[k]==1){
        disease1=disease1+1
    }
}

if(allel.1[k]==1&&allel.2[k]==1){
    marker[k]=2
    count2=count2+1
    if(affectstat[k]==1){
        disease2=disease2+1
    }
}

sumaffect=affectstat[k]+sumaffect

j=j+1

#check
affected=sum(affectstat==1)/c
average=(sum(allel.1==2)+sum(allel.2==2))/(2*c)
pedigree=data.frame(familyID=familyID,childID=childID,fatherID=fatherID,
motherID=motherID,sex=sex,affectstat=affectstat,allel.1=allel.1,allel.2=allel.2,
marker=marker,count0=count0,count1=count1,count2=count2,disease0=disease0,
disease1=disease1,disease2=disease2)
list(familyID=familyID,childID=childID,fatherID=fatherID,motherID=motherID,
sex=sex,affectstat=affectstat,allele.1=allele.1,allele.2=allele.2,marker=marker,
count0=count0,count1=count1,count2=count2,disease0=disease0,disease1=disease1,
disease2=disease2)
}
x=pedisimu(50,3,0.1,0.01,0.5,"add",verbose=TRUE)
# EFBAT: Add "ml" to top line and delete off last blank line
X=cbind(x$familyID,x$childID,x$fatherID,x$motherID,x$sex,
x$affectstat+1,x$allele.1,x$allele.2)
row=(c("ml",rep(NA,7)))
Xnew=t(cbind(row,t(X)))
#rownames(Xnew)=rep("",nrow(Xnew))
pedFBAT=write.table(Xnew,file="ex1.ped",row.names=FALSE,col.names=FALSE,na="")
# Trend Test: Add number of clusters/families to top line and change in "clustexamp.danny.2.c"
X=cbind(x$marker,x$affectstat,1)
row=c(50,rep(NA,2))
Xnew=t(cbind(row,t(X)))
rownames(Xnew)=rep("",nrow(Xnew))
pedTrend=write.table(Xnew,file="ex1.out",row.names=FALSE,col.names=FALSE,na="")
# Conditional Logistic Regression:
pedLog=write.table(cbind(x$familyID,x$childID,x$affectstat,x$marker),file="ex1.txt",
row.names=FALSE,col.names=FALSE)
Appendix II. SAS Script for CLR

ods csv body="C:/Users/lundgren/Research/Logistic/Logistic.csv";
data ex1;
infile "c:\Users\lundgren\Research\Simulation\ex1.txt" dlm=' ' firstobs=1;
input familyID childID affectStat marker;
run;

proc logistic data=ex1 descending;
model affectStat = marker;
strata familyID;
exact marker / estimate = both;
run;

ods csv close;
Appendix III. C Code for QEM

#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#include <conio.h>

#include <limits.h>

#include "nrutil.h"

#include "numrout.c"

#include "nr.h"

#include "trend2.c"

#include "trend3.danny.c"

// #include "score.c"

#define INFILE "ex1.out" // temporarily getting filename from arguments...

#define NUMCLUST 50
#define ALPHA -2.0
#define DELTA 0.1
#define MCSEED 99999999

int main(int argc, char* argv[]){
  int dose[NUMCLUST], litter[NUMCLUST], i, rowm, obscorr, obsstat;
  int sampsz, icount, ncol, numclust, junk;
  int x, y, n, cval, ierr, num, yij[NUMCLUST];
  long mcseed=MCSEED;
  double pval, mpval, xtilde=0, muhat=0, numer=0, den=0, sigma=0, xdev=0, geestat;
  double alpha, sumxdev=0, rhobat, den2, mrstat;

  FILE *fin;
  char INFILE[100];
  if(argc != 2)
    {
      printf(stderr, "usage: %s inputFilename\n", argv[0]);
      exit(1);
    }

  strcpy(INFILE, argv[1]); //get the filename of input file.

  //double mrscore(int numclust, int y[], int clsizes[], int dose[],
//double alpha, double delta, int rowm, int obscorr, int obsstat);
rowm=0;
obscorr=0;
obstat=0;
sampsze=0;
ierr=0;
pval=0;
cval=0;
icount=0;
alpha=0.05;
ncol=NUMCLUST;
fin=fopen(INFILE,"r");
//fin=stdin;
if (fin!=NULL){
    fscanf(fin,"%d",&numclust);
    for(i=0;i<numclust;i++){
        num=fscanf(fin,"%d %d %d",&x,&y,&n);
        //num=fscanf(fin,"%d %d %d",&junk,&x,&y,&n);
        dose[i]=x;
yij[i]=y;
obscorr+=y*(n-y);
sampsze+=n;
rowm+=y;
litter[i]=n;
xtilde+=(n*x);
    }
fclose(fin);
muhat=(float)rowm/(float)sampsze;
xtilde=xtilde/(float)sampsze;
numer=pow(((float)obsstat-(rowm*xtilde)),2);
for(i=0;i<numclust;i++){
    sigma=pow(((float)yij[i]-litter[i]*muhat),2);
xdev=pow(((float)dose[i]-xtilde),2);
sumxdev+=xdev;
den=xdev*sigma;
}
rhohat=((float)rowm/(float)numclust)-((float)rowm*rowm/(numclust*numclust))
+2*(float)obscorr/(float)numclust;
den2=rhohat*sumxdev;
geestat=numer/den;

//mrstat=mr_score(numclust,yij,litter,dose,ALPHA,DELTA,rowm,obscorr,obsstat);
printf("rowm,%d\nsampsz,%d\nobscorr,%d\nobsstat,%d\ngeestat,%7.4f\nmrstat,%7.4f
",
rowm,sampsz,obscorr,obsstat,geestat,mrstat);
trstat(&ncol,litter,dose,&sampsz,&rowm,&obsstat,&obscorr,&alpha,&cval,
&pval,&mpval,&icount,&mcseed,&ierr);
printf("pval,%6.4f\nmonte,%6.4f\ngees stat,%8.4f\nscore test stat,%8.4f
",
pval,mpval,geestat,mrstat);
}
else
{
char * name = INFILE;
fprintf(stderr,"ERROR: can't open file %s\n",name);
}

return 0;
}
Appendix IV. Perl Code to Run in Batch Mode

#!/usr/bin/perl

# runAllSimulations.pl - script to run all the simulations...

# Global Variables:
#$RESEARCH_DIR = "/cygdrive/c/Users/Abbie/Documents/Research";
$RESEARCH_DIR = "/cygdrive/c/Users/lundgren/Research";
#$RESEARCH_DIR = "/cygdrive/c/Users/dperry/Documents/school/helpAbbie/Research/Research";
#$RESEARCH_DIR_WIN = "c:\Users\Abbie\Documents\Research";
$RESEARCH_DIR_WIN = "c:\Users\lundgren\Research";
$SIMULATION_DIR = "$RESEARCH_DIR/Simulation";

# SAS and R paths:
$SAS_EXE = "/cygdrive/c/Users/lundgren/local/SASbin/sas.exe";
$R_EXE = "/cygdrive/c/Users/lundgren/local/Rbin/R.exe";

sub trim($)
{
  my $string = shift;
  $string =~ s/-\s+//;
  $string =~ s/\s+$//;
  return $string;
}

sub runSimulation()
{
  $output = "${SIMULATION_DIR}/runSimulation.sh";
  if(length($output) > 1)
  {
    print "Simulation had output: $output\n";
  }
}

sub runXFBAT()
{
  $XFBAT="${RESEARCH_DIR}/XFBAT/XFBATp.exe";
  $file="${SIMULATION_DIR}/ex1.ped";
  $tmpfile="${RESEARCH_DIR}/XFBAT/tmpex1.ped";
  $tmpfilewin="${RESEARCH_DIR_WIN}\XFBAT\tmpex1.ped";
  $outfile="${RESEARCH_DIR}/XFBAT/tmpout.txt";
  # take out quotes and last blank line:
  }
$sqouput = 'perl ${RESEARCH_DIR}/stripQuotes.pl < $file > $tmpfile';
print $sqouput;
#open pipe to xfbat
#print "| ${XFBAT} > ${outfile}"
open(XCOMMANDS, "| ${XFBAT} > ${outfile}") or die "Could not open xfbat: $!\n";
#print commands to xfbat:
#print XCOMMANDS "$file\n"
print XCOMMANDS "$tmpfilewin\n"
# 7 - analysis
print XCOMMANDS "7\n"
# 8 - quit
print XCOMMANDS "8\n"
# end commands
close(XCOMMANDS);
# read output of XFBAT:
$stat = 0;
$expectation = 0;
$variance = 0;
$pval = 0;
$z = 0;
$inf = 0;
open(XOUTPUT, "$outfile");
while($line = <XOUTPUT>)
{
    #print "xfbat out: $line"
    #the analysis output starts with the word "Marker"
    chomp($line);
    if($line =~ m/Marker/)  
    {
        #Analysis consists of 2 lines, get the 2nd as well:
        $line2 = <XOUTPUT>;
        chomp($line2);
        #Parse the output of the analysis:
        @pairs = split(",
            "$line");
        @pairs2 = split(",
            "$line2");
        @tmp = split(":
            "$pairs2[2]);
        $inf = trim($tmp[1]);
        @tmp = split(":
            "$pairs[2]);
$stat = trim($tmp[1]);
@tmp = split(" ", $pairs2[1]);
$expectation = trim($tmp[1]);
@tmp = split(" ", $pairs2[2]);
$variance = trim($tmp[1]);
@tmp = split(" ", $pairs2[3]);
$pval = trim($tmp[1]);
}
}
if ($pval == "****")
{
  $pval = 1;
}
# computing the asymptotic z-statistic
if ($variance != 0)
{
  $z = ($stat - $expectation) / (sqrt ($variance));
}
else
{
  $z = 0;
}
# return the results in an array:
$result = ($pval, $z, $inf);
return $result;
}
sub runTrend()
{
  $TREND_DIR = "${RESEARCH_DIR}/Trend";
  $TREND_EXE = "${TREND_DIR}/cluster.exe";
  $pval = 0;
  $score = 0;
  %TrendData = {};
  @results = '$TREND_EXE $SIMULATION_DIR/ex1.out';
  foreach $line (@results)
  {
    # take off the endline char
    chomp($line);
# split the line up by comma
@lineparts = split(",",$line);
# now store them in the hash table:
$TrendData{[/lineparts[0]]} = $lineparts[1];
}
$dataName = "pval";
$pval = $TrendData{[/dataName]};
$dataName = "mrstat";
$mrstat = $TrendData{[/dataName]};
# return the results in an array:
@result = ($pval, $mrstat);
return @result;
}
sub runSAS()
{
$LOGISTIC_DIR = "${RESEARCH_DIR}/Logistic";
$OUTPUT = "${LOGISTIC_DIR}/Logistic.csv";
# runLogistic.sh creates 2 files in the Logistic folder: Logistic.log Logistic.csv
$output= '${LOGISTIC_DIR}/runLogistic.sh';
if(length($output) > 1)
{
print "runLogistic had output: $output\n";
}
$SASData = {};
open(SASOUT, "$OUTPUT");
while(my $line = <SASOUT>)
{
if($line =~ m/Response Profile/)
{
do
{
my @parts = split(",",$line);
if($#parts < 0)
{
last;
}
my $lookupName = @parts[0];
#print "lookupName=${lookupName}\n";
my @contents;
$#contents = 0;
while(my $line = <SASOUT>)
{
  if(length($line)<3)
  {
    #print "done appending."
    last;
  }
  #print length($line),": $line"
  chop($line);
  chop($line);
  my @lineparts = split("",$line);
  foreach $p (@lineparts)
  {
    #print "about to append $p 
"
    $contents[$#contents++] = $p;
  }
  #print "lookupName=${lookupName}\n"
  #print "$lookupName : ", join(" ",@contents), "\n"
  $SASData{$lookupName} = [@contents];
} while($line = <SASOUT>)
}
unlink("$OUTPUT");
$title = "\"Testing Global Null Hypothesis: BETA=0\"";
$subElement = 7;
$likelihood = $SASData{$title}{$subElement};
if($likelihood =~ /</)
{
  $likelihood = substr($likelihood,2,length($likelihood)-3);
}
$title = "\"Testing Global Null Hypothesis: BETA=0\"";
$subElement = 11;
$score = $SASData{$title}{$subElement};
if($score == "\n")
{
    $score = substr($score, 2, length($score) - 3);
}
$title = "\"Conditional Exact Tests\"";
$subElement = 10;
$exactPVal = $SASData{$title}[$subElement];
if($exactPVal == "\n")
{
    $exactPVal = substr($exactPVal, 2, length($exactPVal) - 3);
}
if($exactPVal == "." )
{
    $exactPVal = 1;
}
$title = "\"Conditional Exact Tests\"";
$subElement = 11;
$exactMidPVal = $SASData{$title}[$subElement];
if($exactMidPVal == "\n")
{
    $exactMidPVal = substr($exactMidPVal, 2, length($exactMidPVal) - 3);
}
if($exactMidPVal == "." )
{
    $exactMidPVal = 1;
}
# return the values in an array:
$result = ($likelihood, $score, $exactPVal, $exactMidPVal);
return $result;
#
# Main entry point.
sub main()
{
    # This will create 3 files in ${RESEARCH_DIR}/Simulation,
    "exl.ped", "exl.out", "exl.txt"
    print "Running simulation in R...\n";
    runSimulation();
    print "\nProcessing sim results in XFBAT...\n";
$xfbatResults = runXFBAT();
print "stat: $xfbatResults[0]\n";
print "exp: $xfbatResults[1]\n";
print "var: $xfbatResults[2]\n";
print "pval: $xfbatResults[3]\n";
print "z: $xfbatResults[4]\n";
print "Processing sim results in cluster.exe...\n";
trendResults = runTrend();
print "pval: $trendResults[0]\n";
print "mrstat: $trendResults[1]\n";
print "Processing sim results in SAS ...
";
&SASResults = runSAS();
print "likelihood: $SASResults[0]\n";
print "score: $SASResults[1]\n";
print "exact p-val: $SASResults[2]\n";
print "exact mid p-val: $SASResults[3]\n";
}
# Main entry point.
sub mainLooped()
{
    $iterations = 10000;
    $xfbatPvalTotal = 0;
    $xfbatPvalMax = 0;
    $xfbatPvalMin = 2;
    $xfbatPvalCount = 0;
    $xfbatZCount = 0;
    $trendPvalTotal = 0;
    $trendPvalMax = 0;
    $trendPvalMin = 2;
    $trendPvalCount = 0;
    $trendMrstatCount = 0;
    $SASPvalTotal = 0;
    $SASMidPvalTotal = 0;
    $SASPvalMax = 0;
    $SASPvalMin = 2;
    $SASMidPvalCount = 0;
    $SASLikelihoodCount = 0;
```bash
$SASScoreCount = 0;
for($i=0 ; $i<$iterations ; $i++)
{
    #for a ghetto progress bar, print a period for each iteration..
    print ";
    # This will create 3 files in ${RESEARCH_DIR}/Simulation, "ex1.ped", "ex1.out", "ex1.txt"
    runSimulation();
    @xfbatResults = runXFBAT();
    $xfbatPvalTotal += $xfbatResults[0];
    if($xfbatResults[0] < $xfbatPvalMin)
    {
        $xfbatPvalMin = $xfbatResults[0];
    }
    if($xfbatResults[0] > $xfbatPvalMax)
    {
        $xfbatPvalMax = $xfbatResults[0];
    }
    if($xfbatResults[0] < .05)
    {
        ++$xfbatPvalCount;
    }
    if(($xfbatResults[1] > 1.96) || ($xfbatResults[1] < - 1.96))
    {
        ++$xfbatZCount;
    }
    @trendResults = runTrend();
    $trendPvalTotal += $trendResults[0];
    if($trendResults[0] < $trendPvalMin)
    {
        $trendPvalMin = $trendResults[0];
    }
    if($trendResults[0] > $trendPvalMax)
    {
        $trendPvalMax = $trendResults[0];
    }
    if($trendResults[0] < .05)
    {
    }
}
++$trendPvalCount;
}
if($trendResults[1] < .05)
{
  ++$trendMrstatCount;
}
$SASResults = runSAS();
$SASPvalTotal += $SASResults[2];
$SASMidPvalTotal += $SASResults[3];
if($SASResults[2] < $SASPvalMin)
{
  $SASPvalMin = $SASResults[2];
}
if($SASResults[2] > $SASPvalMax)
{
  $SASPvalMax = $SASResults[2];
}
if($SASResults[2] < .05)
{
  ++$SASPvalCount;
}
if($SASResults[3] < .05)
{
  ++$SASMidPvalCount;
}
if($SASResults[0] < .05)
{
  ++$SASLikelihoodCount;
}
if($SASResults[1] < .05)
{
  ++$SASScoreCount;
}
#now average them
$xfbatPvalAvg = $xfbatPvalTotal / ($iterations * 1.0);
$xfbatPvalPower = $xfbatPvalCount/ ($iterations * 1.0);
$xfbatZPower = $xfbatZCount / ($iterations * 1.0);
```bash
$trendPvalAvg = $trendPvalTotal / ($iterations * 1.0);
$trendPvalPower = $trendPvalCount / ($iterations * 1.0);
$SASPvalAvg = $SASPvalTotal / ($iterations * 1.0);
$SASPvalPower = $SASPvalCount / ($iterations * 1.0);
$SASMidPvalAvg = $SASMidPvalTotal / ($iterations * 1.0);
$SASMidPvalPower = $SASMidPvalCount / ($iterations * 1.0);
$SASLikelihoodPower = $SASLikelihoodCount / ($iterations * 1.0);
$SASScorePower = $SASScoreCount / ($iterations * 1.0);

# now print the results:
print "\n";
print "xfbat p-val
Min : $xfbatPvalMin
Avg : $xfbatPvalAvg
Max : $xfbatPvalMax
PvalPower : $xfbatPvalPower
ZPower : $xfbatZPower\n\n";
print "trend p-val
Min : $trendPvalMin
Avg : $trendPvalAvg
Max : $trendPvalMax
PvalPower : $trendPvalPower\n\n";
print "sas p-val
Min : $SASPvalMin
Avg : $SASPvalAvg
Max : $SASPvalMax
PvalPower : $SASPvalPower
MidPval : $SASMidPvalAvg
MidPvalPower : $SASMidPvalPower
LikelihoodPower : $SASLikelihoodPower
ScorePower : $SASScorePower\n\n";
}
#main();
mainLooped();
```