Original Paper

Human Heredity

Hum Hered 2014;78:179–194
DOI: 10.1159/000369065

Received: June 23, 2014
Accepted after revision: October 14, 2014
Published online: January 29, 2015

Fast Linkage Analysis with MOD Scores Using Algebraic Calculation

Markus Brugger Konstantin Strauch

Institute of Medical Informatics, Biometry and Epidemiology, Chair of Genetic Epidemiology, Ludwig-Maximilians-Universität, Munich, and Institute of Genetic Epidemiology, Helmholtz Zentrum München – German Research Center for Environmental Health, Neuherberg, Germany

Key Words
Parametric linkage analysis · MOD scores · Lander-Green algorithm · Algebraic calculation · Rare variants

Abstract

Objective: As the mode of inheritance is often unknown for complex diseases, a MOD-score analysis, in which the parametric LOD score is maximized with respect to the trait-model parameters, can be a powerful approach in genetic linkage analysis. Because the calculation of the disease-locus likelihood is the most time-consuming step in a MOD-score analysis, we aimed to optimize this part of the calculation to speed up linkage analysis using the GENEHUNTER-MODSCORE software package. ***Methods:*** Our new algorithm is based on minimizing the effective number of inheritance vectors by collapsing them into classes. To this end, the disease-locus-likelihood contribution of each inheritance vector is represented and stored in its algebraic form as a symbolic sum of products of penetrances and disease-allele frequencies. Simulations were used to assess the speedup of our new algorithm. ***Results:*** We were able to achieve speedups ranging from 1.94 to 11.52 compared to the original GENEHUNTER-MODSCORE version, with higher speedups for larger pedigrees. When calculating p values, the speedup ranged from 1.69 to 10.36. ***Conclusion:*** Computation times for MOD-score analysis, involving the evaluation of many tested sets of trait-model parameters and p value calculation, have been prohibitively high so far. With our new algebraic algorithm, such an analysis is now feasible within a reasonable amount of time.

Introduction

Since its first successful application by the physician and geneticist Jan Mohr in 1954 [1], linkage analysis has been a powerful tool in human disease gene mapping for many decades. With this method, many Mendelian disease genes have been mapped to their genetic loci by the use of family data [2]. Due to the development of genotyping techniques with dense SNP marker panels and the progressing availability of large case-control or population-based cohorts, association analysis has recently become the preferred method for statistical analysis in the field of genetic epidemiology. Unlike linkage analysis, an association analysis can make use of samples with unrelated individuals; it does not require families which are obviously much harder to recruit. However, with the advent of next-generation sequencing data and increasing interest in the analysis of rare variants, the analysis of family data using linkage analysis is undergoing a renaissance. The basis for this interest is that numerous rare

KARGER

E-Mail karger@karger.com
www.karger.com/hhe

Markus Brugger
Institute of Genetic Epidemiology, Helmholtz Zentrum München
German Research Center for Environmental Health
Ingolstädter Landstrasse 1, DE–85764 Neuherberg (Germany)
E-Mail markus.brugger@helmholtz-muenchen.de

variants with moderate effects may explain an appreciable amount of the missing heritability [3]. Although rare variants are individually rare, a single person can have thousands of such rare variants across the genome. It can thus be difficult to determine whether the observation of a rare variant is a sequencing artifact or in fact a true variant if it is carried by only a single individual of the sample. However, one expects that rare variants segregate and accumulate within families. Results from the Genetic Analysis Workshop 17 showed that analyses using whole-exome sequencing data require much smaller sample sizes when working with families than with unrelated individuals, because the ability to detect rare causal variants is enhanced in family studies as the variants are carried by several family members jointly [4].

In parametric linkage analysis, which is also known as LOD-score or model-based analysis, a certain set of trait-model parameters is explicitly assumed for the segregation of the disease. In the simplest case of a diallelic autosomal trait locus, which is assumed throughout this paper, these parameters are the disease-allele frequency p and the three penetrances f_0, f_1, and f_2, with f_i denoting the probability that an individual with i copies of the disease allele is affected by the disease. The central part of parametric linkage analysis is the computation of the genetic likelihood, which is based on the following parameters: disease-allele frequency, penetrances, marker-allele frequencies, and the recombination fractions – and, if applicable, linkage disequilibria between the loci. In addition, the relation between family members is required to be known. Eventually, a likelihood-ratio test is performed, in which the likelihood under the alternative hypothesis of linkage with some specific value of the recombination fraction ($\theta < 0.5$; the numerator of the likelihood ratio) is compared to the null hypothesis of no linkage ($\theta = 0.5$; the denominator of the likelihood ratio). The logarithm to the base 10 of this likelihood ratio is the LOD score [5]. It is maximized by varying θ between marker and trait locus in the numerator (maximum LOD score). Trait-model parameters can either be prespecified according to results from previous segregation analyses or maximized along with the recombination fraction in a joint segregation and linkage analysis. The latter approach is also known as MOD-score analysis and has been first proposed by Risch [6]. As the power of a LOD-score analysis crucially depends on the true mode of inheritance, which is generally unknown, a MOD-score analysis can have greater power to detect linkage than a simple LOD-score analysis. Furthermore, in case of a trait-model-parameter misspecification, the recombination fraction will be overestimated [7]. In a multipoint analysis, the misspecification may even lead to an exclusion of linkage [8]. Simulations have shown that, especially when analyzing a mixture of different types of pedigrees, the MOD-score approach outperforms other linkage methods in terms of power to identify genes with modest effect [9]. Due to the maximization over trait-model parameters, MOD scores are inflated when compared to LOD scores. Since the asymptotic distribution of MOD scores is unknown in the general case, p values for the linkage test must be obtained by simulating the distribution of the MOD score under the null hypothesis of no linkage. Our group has implemented the MOD-score approach, including a routine to perform simulations under the null hypothesis, in the GENEHUNTER-MODSCORE (GHM) software [10–13]. Its application has led to the identification of a variety of genetic disease loci [14–18].

Nonparametric linkage methods have been proposed in order to avoid trait-model misspecification that occurs when using simple LOD-score analyses. These methods test if affected pedigree members have more alleles in common than would be expected by chance under the null hypothesis of no linkage. Nonparametric methods are often considered to be 'model-free' because they do not rely on explicit assumptions as to the trait-model parameters. However, Knapp et al. [19] have shown that, for samples of affected sib pairs (ASPs) with the parents' phenotypes unknown or set to unknown, the nonparametric mean test is equivalent to a LOD-score analysis under a recessive mode of inheritance, and the possible triangle test proposed by Holmans [20] is equivalent to a MOD-score analysis. In the possible triangle test, the genetic likelihood is expressed in terms of the probabilities z_0, z_1, and z_2 that an ASP shares 0, 1, or 2 alleles identical-by-descent (IBD) with restrictions to genetically possible models [20]. These allele-sharing probabilities can be expressed as functions of the trait-model parameters f_0, f_1, f_2, p, and θ [21], and hence, the parametric and nonparametric likelihood are identical. More generally, the allele-sharing probabilities of any pedigree with affected relatives could be used to construct a nonparametric allele-sharing-based test statistic [22]. However, for such a nonparametric test to be constructed for a certain pedigree type other than ASPs or affected half-sib pairs (AHSPs) would yet demand knowledge as to how many allele-sharing classes exist for that pedigree type and how the corresponding restrictions to genetically possible models can be formulated. Knapp [23] derived allele-sharing probabilities for affected sib triplets (ASTs) with parental phenotypes set to unknown. However, the restrictions to genetically possible models cannot be expressed in closed form. But again, the allele-sharing probabilities, which represent the truly underlying parameters, can be modeled as a function of f_0, f_1, f_2, p, and θ. Hence, the parametric and nonparametric likelihood are identical even beyond the special cases of ASPs and AHSPs, and MOD-score analysis is equivalent to the likelihood-ratio test based on allele-sharing parameters. As outlined by Strauch [22], this holds for any type of pedigree.

The calculation of the genetic likelihood is pivotal for both parametric and nonparametric linkage analysis. Given the complexity of real family data, it cannot be calculated manually in most cases. Large pedigrees, many markers, and missing genotypes lead to a substantial number of possible genotype combinations that must be considered in the likelihood. Two major algorithms are known that allow for the calculation of the likelihood: the Elston-Stewart [24] and the Lander-Green algorithm [25]. The former is genotype-oriented and is based on the peeling of nuclear families. It makes use of the independence of genotypes of different nuclear families within a pedigree when conditioning on a certain genotype of the connecting person, the so-called pivot. The Elston-Stewart algorithm thereby summarizes identical terms that correspond to a particular genotype combination within the likelihood. The algorithm scales linearly with the number of individuals in a pedigree and exponentially with the number of analyzed loci. Hence, it is limited to the analysis of a relatively small number of genetic markers. The Elston-Stewart algorithm has been implemented and further optimized in several linkage software packages such as LINKAGE [26–28], FASTLINK [29, 30], VITESSE [31, 32], and PSEUDOMARKER [33, 34]. The Lander-Green algorithm is complementary to the Elston-Stewart algorithm, such that it treats each marker locus one after another and distinguishes the marker loci from the disease locus. The Lander-Green algorithm is implemented in several genetic analysis software packages such as GENEHUNTER [35], ALLEGRO [36, 37], and MERLIN [38]. It scales linearly with the number of markers and exponentially with the number of individuals in a pedigree. Therefore, the Lander-Green algorithm is well suited for the analysis of large datasets of genetic markers, which are typically available for small to moderately large pedigrees when mapping complex-disease genes. In addition, it allows both parametric and nonparametric linkage analysis. This is because, as a first step, inheritance information is extracted solely from marker data by applying the concept of inheritance vectors. Then, a parametric or nonparametric scoring function that incorporates information with regard to the disease phenotypes of the pedigree members is applied to evaluate a set of genetic positions of the putative trait locus in terms of linkage with the markers. In the parametric case, the scoring function corresponds to the ratio of the disease-locus likelihoods under the assumption of linkage versus no linkage.

In this paper, we describe a new algorithm for the calculation of the parametric disease-locus likelihood in the context of the Lander-Green algorithm. This part of the calculation is the most time-consuming step in a MOD-score analysis. How can it be accelerated? Our new approach to a faster implementation is structured according to the following three aspects:

- *Inheritance Vectors and the Identity of the MOD Score with the Allele-Sharing-Based Test Statistic.* Inspired by the identity of the allele-sharing-based nonparametric likelihood and the parametric likelihood in the test for linkage, our new algorithm is based on minimizing the effective number of inheritance vectors by collapsing them into classes, whose members are observed with the same probability function of f_0, f_1, f_2, and p, i.e. having the same allele-sharing proportions for a given type of pedigree structure. This approach has the potential to considerably reduce the number of floating number operations, because instead of calculating the disease-locus-likelihood contribution for a given set of trait-model parameters for each inheritance vector, it needs to be calculated only once for all members of a certain class.
- *Algebraic Formulation of the Disease-Locus Likelihood.* To collapse inheritance vectors into certain classes, i.e. to recognize which vectors belong to the same class, the disease-locus-likelihood contribution of each inheritance vector must be represented and stored in its algebraic form. This involves representing it as a symbolic sum of products of penetrances and disease-allele frequencies for a given combination of disease-locus genotypes of all individuals in the pedigree. Inheritance vectors with identical symbolic sums can thus readily be grouped into the same class. This step involves no numerical calculation and needs to be done only once at the beginning of a MOD-score analysis for a given pedigree.
- *Exploiting Similarities in Family Structures by the Use of Inheritance Vector Classes.* Two pedigrees with a certain pattern of disease status, each of which can be represented by a directed acyclic graph, are indistin-

guishable in terms of the disease-locus-likelihood structure if they are comprised of the same set of inheritance vector classes and the same number of vector members per class. Hence, two such pedigrees yield the same disease-locus-likelihood contributions. The computational effort for LOD-score calculation for the second pedigree can be entirely avoided. When two pedigrees are distinct, i.e. yielding different sets of inheritance vector classes, identical symbolic products are still stored in a common database to avoid dispensable numerical calculations. The computational effort during the LOD-score calculation is hence further reduced by the degree of similarity of pedigrees based on their inheritance vector classes.

In conjunction with the already existing options and optimizations of GHM, which are addressed below, our new algorithm allows for a rapid evaluation of the likelihood for a large number of disease models, as required during maximization over trait models in a MOD-score analysis. The reduction of computing time is a prerequisite for empirically determining p values by performing simulations and MOD-score calculations of many replicates.

It has to be noted that the first version of GHM [13] is based on GENEHUNTER version 2.1 [39]. Since the release of GENEHUNTER version 1.0 in 1996 [35], many improvements have been implemented, which have led to a significant analysis speedup and which have added various additional functionalities to the software package [39–41]. However, these previous improvements did not concern the calculation of the parametric disease-locus likelihood as does our new algebraic algorithm. All improvements as of GENEHUNTER version 2.1 [39] have been carried forward to GHM and are complementary to the algebraic algorithm presented in this paper. For more information on the original GENEHUNTER software, we refer to the review by Nyholt [42].

Methods

The Lander-Green Algorithm

Inheritance Vectors

As a first step, the Lander-Green algorithm enumerates all possible inheritance vectors in a pedigree. An inheritance vector denotes a possible family-specific pattern of segregation of founder alleles. Each bit of the inheritance vector corresponds to the outcome of a certain meiosis, which codes the transmission of the grand-paternally or grand-maternally inherited allele to the child as a value of 0 or 1, respectively. With n non-founders, there are $2n$ meioses and 2^{2n} possible inheritance vectors. However, even if the information is complete, there are 2^f remaining inheritance vectors that all have the same probability. This is due to the fact that the parental origin of founder haplotypes is unknown. In other words, the bit corresponding to the first child of each founder can be fixed arbitrarily (e.g. to a value of 0). Hence, the 2^{2n} inheritance vectors can be grouped into $2^{(2n-f)}$ equivalence classes, each comprising 2^f inheritance vectors.

Probability of Observed Marker Genotypes Given a Particular Inheritance Vector

The algorithm iterates over inheritance vectors and markers and calculates the probability of the observed genotypes for each marker conditional on a particular inheritance vector [25]. This step of the calculation is based on a graph-theoretical process. Following the notation in Kruglyak et al. [35], let $G(v)$ be a graph for a given inheritance vector v whose vertices are the founder alleles $X = \{x_1, x_2, \ldots, x_{2f}\}$ corresponding to the $2f$ founder alleles at the marker locus, which are assumed to be distinct by descent ('placeholder alleles'). An inheritance vector v specifies the placeholder alleles inherited by each individual in the pedigree. The lines connecting the two placeholder alleles that correspond to the genotype of each individual, as defined by the inheritance vector v, represent the edges of the graph. The placeholder alleles are then assigned the actual founder alleles at the marker locus, and placeholder allele assignments that are incompatible with the observed marker genotypes are eliminated from further consideration. Then, the probability of drawing the founder alleles from the population, i.e. the product of allele frequencies of all founders, is calculated, and the sum of this product is taken over all possible founder allele assignments that are compatible with both the inheritance vector and the observed marker genotypes.

The Markov Chain

The Lander-Green algorithm uses a Markov process to describe the joint distribution of inheritance vectors along a chromosome [25]. This is based on the observation that, under the assumption of no genetic interference, inheritance vectors form a hidden Markov chain. The observed states are the typed marker genotypes, and the hidden states are the inheritance vectors. The matrices of transition probabilities between inheritance vectors at consecutive markers are a function of recombination fractions between markers. After the inheritance vector distribution ($P_{complete}$) has been calculated at a certain genetic position, the disease phenotypes of the family members are considered by using an appropriate scoring function.

The Scoring Function

At this stage of the analysis, different scoring functions are defined for parametric and nonparametric linkage analysis. In a parametric analysis, the scoring function is the ratio of the disease-locus likelihoods under linkage in the nominator versus under no linkage in the denominator. The disease-locus likelihood is calculated conditional on each inheritance vector. As marker information is often incomplete, several inheritance vectors are possible, and the conditional probabilities of these vectors given the marker information ($P_{complete}$) have a nonzero value. Therefore, the sum of the scoring function is taken over all inheritance vectors weighted by their conditional probability given the marker information ($P_{complete}$). Under no linkage between marker and disease locus, the probability of each inheritance vector no longer depends on the marker data. Hence, the inheritance vector distribution at a putative disease lo-

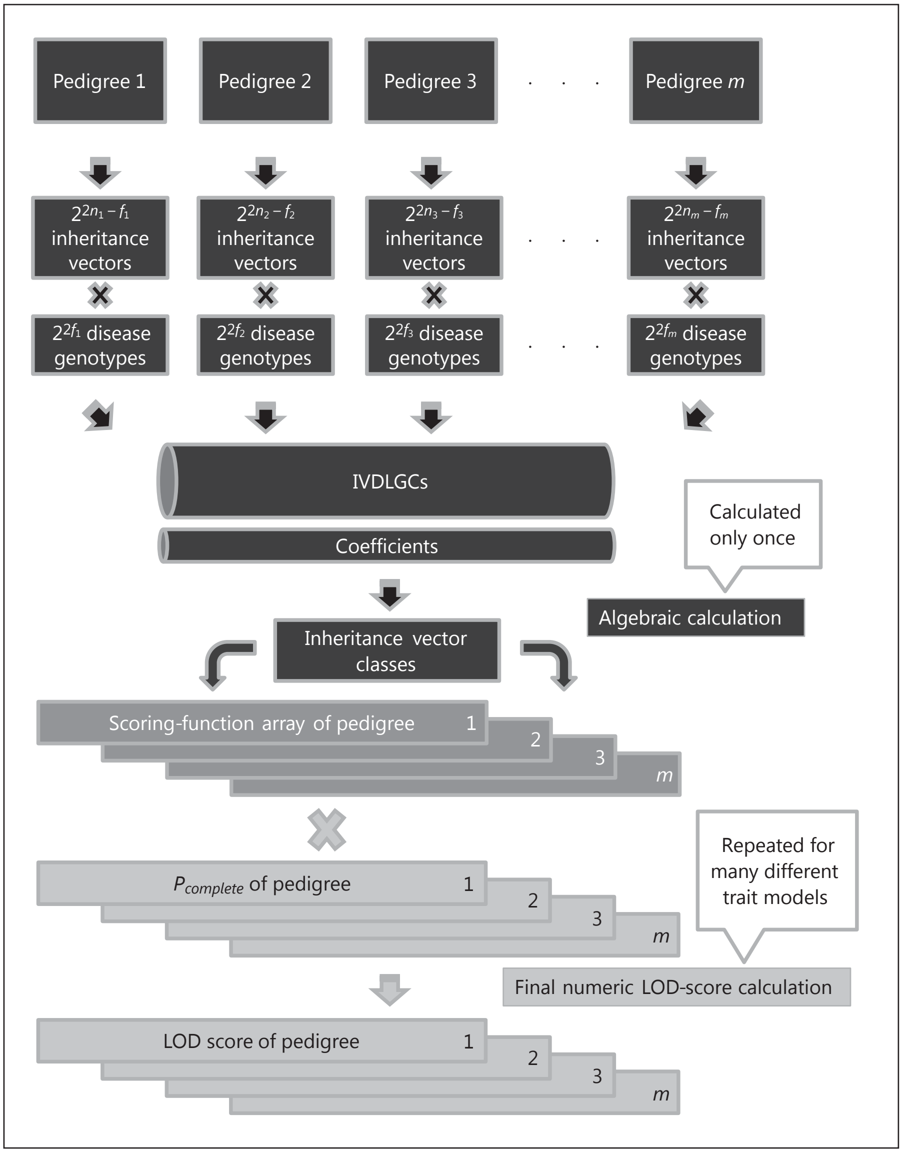

Pedigree 1
Pedigree 2
Pedigree 3
Pedigree m
$2^{2n_1 - f_1}$ inheritance vectors
$2^{2n_2 - f_2}$ inheritance vectors
$2^{2n_3 - f_3}$ inheritance vectors
$2^{2n_m - f_m}$ inheritance vectors
2^{2f_1} disease genotypes
2^{2f_2} disease genotypes
2^{2f_3} disease genotypes
2^{2f_m} disease genotypes
IVDLGCs
Coefficients
Calculated only once
Algebraic calculation
Inheritance vector classes
Scoring-function array of pedigree 1 2 3 m
$P_{complete}$ of pedigree 1 2 3 m
Repeated for many different trait models
Final numeric LOD-score calculation
LOD score of pedigree 1 2 3 m

Fig. 1. Depiction of the algebraic algorithm. Steps that have to be calculated only once are highlighted in black. The final LOD-score calculation is shaded in light grey and the interface between the algebraic algorithm and the numeric LOD-score calculation – the scoring-function arrays of the pedigrees – is shown in dark grey. Each inheritance vector of a given pedigree with n nonfounders and f founders is analyzed in regard to its disease-locus-likelihood contribution. For a given inheritance vector, all possible disease-locus-genotype combinations must be considered. Each disease-locus-genotype combination yields a likelihood contribution that is a product of penetrances and disease-allele frequencies. The sum over all disease-locus-genotype combinations is the total disease-locus-likelihood contribution of the given inheritance vector. The likelihood contribution of each IVDLGC is stored in its algebraic form. IVDLGCs of a given inheritance vector that lead to the same algebraic representation are joined together by including a coefficient. Inheritance vectors with the same set of IVDLGCs are assigned to a certain inheritance vector class. The analysis of inheritance vectors is performed for all pedigrees of the dataset, whereby all pedigrees of the sample have a joint IVDLGC storage. This way, a certain inheritance vector class can comprise inheritance vectors of several pedigrees. Finally, the trait-model-specific LOD score is calculated numerically as the scalar product of $P_{complete}$ and the scoring-function array. This step is repeated many times during a MOD-score analysis by numerically evaluating the scoring-function arrays assuming different sets of trait-model parameters.

cus position unlinked to the marker locus corresponds to a uniform distribution with probability $1/2^{(2n-f)}$ for each inheritance vector. Maximizing the logarithm to the base 10 of this likelihood ratio over the recombination fraction θ yields the LOD score. When it is maximized over (f_0, f_1, f_2, and p) in addition to θ, the MOD score is obtained. Nonparametric scoring functions count the number of alleles shared IBD by affected pedigree members given a certain inheritance vector. Popular nonparametric scoring functions are S_{pairs} and S_{all} [35, 43]. Our new algorithm only affects the calculation of the parametric scoring function, and we refer to McPeek [44] for more information about nonparametric scoring functions.

The Algebraic Algorithm

Basic Concept

As described by Strauch [22], inheritance vectors can be collapsed into inheritance vector classes if they cannot be distinguished from each other on the basis of the phenotypic structure of a given family tree. In other words, inheritance vectors being observed with the same allele-sharing probability z_i conditional on the disease phenotypes and the parameters f_0, f_1, f_2, and p are comprised in a certain inheritance vector class. The number of inheritance vector classes, and hence allele-sharing probabilities, depends on the number of persons in a pedigree and hence differs between different types of pedigrees in a sample. As stated before, it appears to be very difficult to construct a nonparametric allele-sharing test, which uses the probabilities z_i, along the lines of the possible triangle test for ASPs, for each of the various pedigree types contained in the particular sample under study. In addition, the restriction to genetically possible models is difficult to formulate. However, given the identity of the parametric likelihood with the nonparametric likelihood in an allele-sharing-based test and the consequential fact that the z_is are a function of (f_0, f_1, f_2, and p), it seems straightforward to use the parametric formulation of the disease-locus likelihood and to collapse those inheritance vectors into a certain class that, by an identical probability z_i, lead to the same likelihood contribution. An algorithm that makes use of this structure has the potential to substantially reduce the computational effort involved in the disease-locus-likelihood calculation for a given pedigree, since the likelihood needs to be calculated only for one member of each class.

Analysis of Inheritance Vectors

Our new algorithm starts by analyzing each of the $2^{(2n-f)}$ inheritance vectors of a certain pedigree with regard to its disease-locus-likelihood contribution. The processing of the marker-locus likelihood by the GHM software using hidden Markov models to calculate $P_{complete}$ remains untouched by our new approach. The consecutive steps of the algebraic algorithm can be followed by looking at figure 1, which depicts the analysis of all pedigrees in a dataset. For the present, we assume that there is only a single pedigree in the dataset. For a given inheritance vector, all possible disease-locus-genotype combinations must be considered. Each disease-locus-genotype combination yields a likelihood contribution that is a product of penetrances and disease-allele frequencies. The sum over all disease-locus-genotype combinations is the total disease-locus-likelihood contribution of the given inheritance vector. In order to avoid many floating point operations each time an inheritance-vector-disease-locus-genotype combination (IVDLGC) is considered, every IVDLGC is stored in its algebraic form. This way, each inheritance vector can be considered as a set of a certain number of IVDLGCs, whereby our algorithm builds up a database of IVDLGCs, such that only combinations leading to a new algebraic representation are additionally stored in memory. Essentially, IVDLGCs are stored in a big table and connected to the inheritance vector classes by the use of pointers. Pointers are a powerful feature for memory access specific to the C programming language, in which GHM is written. IVDLGCs of a given inheritance vector that lead to the same algebraic representation, i.e. the product of a certain combination of parameters (f_0, f_1, f_2, and p), are joined together by incrementing a coefficient (integer) and thus need not be saved separately, which avoids extra floating point operations and memory.

Identification of Inheritance Vector Classes

All inheritance vectors of a certain class consist of the same set of IVDLGCs. In particular, if an inheritance vector has the same set of IVDLGCs as an inheritance vector class already identified during the course of the calculation, the vector is added to that class. A previously unobserved set of IVDLGCs for a certain vector leads to the definition of a new inheritance vector class. An inheritance vector class corresponds to a certain allele-sharing class in the nonparametric context. Figure 2 gives a technical depiction of the algebraic algorithm for an AST. It illustrates how a specific inheritance vector is assigned to its corresponding class on the basis of the algebraic calculation of its disease-locus-likelihood contribution.

Calculation of the LOD Score

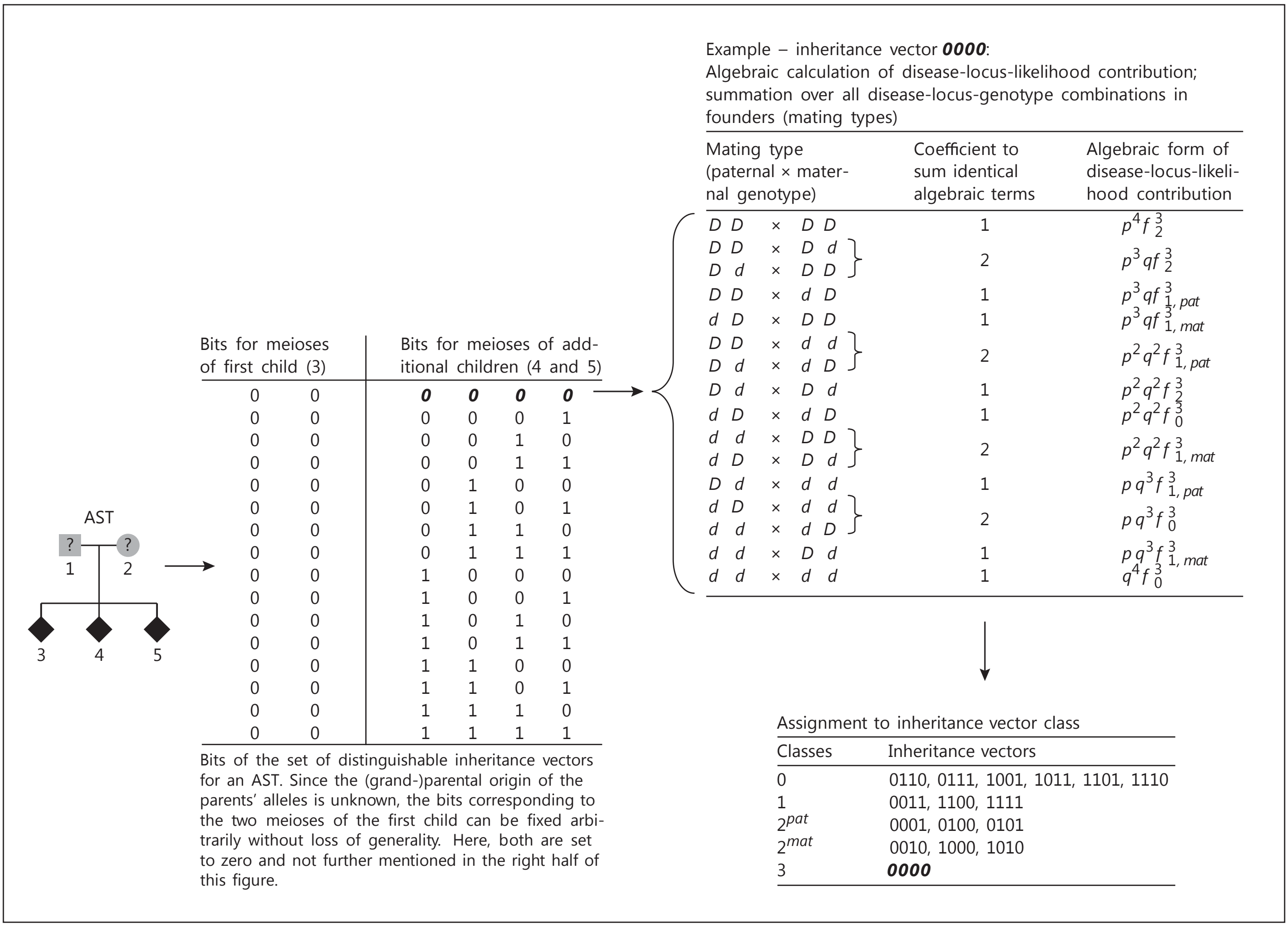
When all inheritance vectors of a given pedigree have been assigned to a certain inheritance vector class and the algebraic structure mentioned above has been determined, the LOD score can readily be calculated for a given set of trait-model parameters. To this end, the algebraic representations of IVDLGCs of all inheritance vector classes are evaluated numerically by inserting the (numeric) values of the parameters (f_0, f_1, f_2, and p) of a specified disease model. The result of each of these products is further multiplied by its associated coefficient, which is equal to the number of IVDLGCs with the same product in a given inheritance vector class, and the sum is taken over all products of that class. This way, the disease-locus-likelihood contributions of all inheritance vector classes are calculated in a single step and then copied into the scoring-function array of the pedigree, according to the class to which a certain inheritance vector belongs. The step of finding the disease-locus-likelihood contribution of the inheritance vector class that corresponds to a given inheritance vector involves the use of pointers and dereference operations. Finally, the trait-model-specific LOD score is calculated as the scalar product of $P_{complete}$ and the scoring-function array. It is of note that information from marker data only affects the calculation of $P_{complete}$, which furthermore is independent of the trait-model parameters. Consequently, $P_{complete}$ has to be computed once for every genetic position and every pedigree in the dataset, even if some or many pedigrees have the same structure. However, $P_{complete}$ can be reused for the LOD-score evaluations under many different trait-model parameters during the maximization.

Number of Inheritance Vector Classes

The degree to which inheritance vectors can be collapsed into certain inheritance vector classes, and hence the computational speedup, depends on the pedigree size and the phenotypes of its members. For example, with nuclear families and parental phenotypes unknown, the potential of reduction by collapsing inheritance vectors into classes increases from ASPs over ASTs to larger sibships. ASPs with 4 possible inheritance vectors have 3 distinct allele-sharing classes, i.e. inheritance vector classes (0, 1, or 2 alleles shared IBD). If imprinting is modeled, e.g. using the four-penetrance formulation developed by Strauch et al. [45] as implemented in GENEHUNTER-IMPRINTING and GHM, ASPs have 4 allele-sharing classes (in this case, the class of 1 shared allele is further distinguished by the parental origin). ASTs with 16 possible inheritance vectors have 4 and 5 allele-sharing classes for a nonimprinting and an imprinting model, respectively (Appendix) [23]. In the following, we will assume an imprinting model when deriving allele-sharing classes, because GHM internally always uses the four-penetrance formulation. The total number of inheritance vectors as well as the reduced number of vector classes are given in table 1 as a function of sibship size of a nuclear family with parental phenotypes unknown (or set to unknown).

Extension across Pedigrees

A further advantage of the algebraic algorithm is that the concept of storing IVDLGCs can even be extended across pedigrees, such that all pedigrees of the sample have a joint IVDLGC storage. A pedigree can thus be considered as a set of certain inheritance vector classes each consisting of a certain set of IVDLGCs. This structure, which is the basis of the algebraic algorithm, is depicted in figure 1. Here, in contrast to the case of considering a single pedigree, a certain inheritance vector class can comprise inheritance vectors of several pedigrees. Hence, the disease-locus-likelihood contributions of all inheritance vector classes are calculated in a single step for the entire dataset, and then the result for a certain inheritance vector class is used for all pedigrees with inheritance vectors that are members of that particular class.



Bits for meioses of first child (3)	Bits for meioses of additional children (4 and 5)
0 0	**0 0 0 0**
0 0	0 0 0 1
0 0	0 0 1 0
0 0	0 0 1 1
0 0	0 1 0 0
0 0	0 1 0 1
0 0	0 1 1 0
0 0	0 1 1 1
0 0	1 0 0 0
0 0	1 0 0 1
0 0	1 0 1 0
0 0	1 0 1 1
0 0	1 1 0 0
0 0	1 1 0 1
0 0	1 1 1 0
0 0	1 1 1 1

Bits of the set of distinguishable inheritance vectors for an AST. Since the (grand-)parental origin of the parents' alleles is unknown, the bits corresponding to the two meioses of the first child can be fixed arbitrarily without loss of generality. Here, both are set to zero and not further mentioned in the right half of this figure.

Example – inheritance vector ***0000***:
Algebraic calculation of disease-locus-likelihood contribution; summation over all disease-locus-genotype combinations in founders (mating types)

Mating type (paternal × maternal genotype)	Coefficient to sum identical algebraic terms	Algebraic form of disease-locus-likelihood contribution
D D × *D D*	1	$p^4 f_2^3$
D D × *D d*	2	$p^3 q f_2^3$
D d × *D D*		
D D × *d D*	1	$p^3 q f_{1, pat}^3$
d D × *D D*	1	$p^3 q f_{1, mat}^3$
D D × *d d*	2	$p^2 q^2 f_{1, pat}^3$
D d × *d D*		
D d × *D d*	1	$p^2 q^2 f_2^3$
d D × *d D*	1	$p^2 q^2 f_0^3$
d d × *D D*	2	$p^2 q^2 f_{1, mat}^3$
d D × *D d*		
D d × *d d*	1	$p q^3 f_{1, pat}^3$
d D × *d d*	2	$p q^3 f_0^3$
d d × *d D*		
d d × *D d*	1	$p q^3 f_{1, mat}^3$
d d × *d d*	1	$q^4 f_0^3$

Assignment to inheritance vector class

Classes	Inheritance vectors
0	0110, 0111, 1001, 1011, 1101, 1110
1	0011, 1100, 1111
2^{pat}	0001, 0100, 0101
2^{mat}	0010, 1000, 1010
3	***0000***

Fig. 2. Technical depiction of the algebraic algorithm for an AST. If several inheritance vectors have the same disease-locus-likelihood contribution, they are joined together in an inheritance vector class.

Table 1. Allele-sharing classes for affected sibships

	ASP	AST	ASQ	ASQui	ASS
Number of inheritance vectors ($2^{(2n-f)}$)	4	16	64	256	1,024
Inheritance vector classes with imprinting taken into account	4	5	11	14	24
Reduction factor ($2^{(2n-f)}$/number of inheritance vector classes)	1	3.2	5.82	18.29	42.67

ASP = Affected sib pair; AST = affected sib triplet; ASQ = affected sib quadruplet; ASQui = affected sib quintet; ASS = affected sib sextet.

SpeedUp

The initial effort of the algebraic algorithm to identify the inheritance vector classes of all pedigrees is high, but the ensuing calculation of LOD scores assuming a large number of disease models is sped up considerably, especially when a dataset is comprised of pedigrees of only a few types. For example, in a dataset of 1,000 ASTs, the disease-locus-likelihood contributions of the 5 inheritance vector classes, given a certain disease model, have to be calculated only once for the whole dataset rather than 1,000 times.

The Peeling Algorithm

In the original version of GHM, the calculation of the parametric disease-locus likelihood is done separately for each inheritance vector by applying the Elston-Stewart algorithm, i.e. peeling nuclear

Table 2. Allele-sharing classes for discordant scenarios

	DSP	DSQ	DML	D3G
Number of inheritance vectors ($2^{(2n-f)}$)	4	64	64	128
Inheritance vector classes with imprinting taken into account	4	28	64	80
Reduction factor ($2^{(2n-f)}$/number of inheritance vector classes)	1	2.29	1	1.6

DSP = Discordant sib pair; DSQ = discordant sib quadruplet; DML = discordant marriage loop; D3G = discordant three-generation pedigree.

Table 3. Overview of scenarios for run-time assessment

Dataset No.	1	2	3	4	5	6	7	8
Pedigree type	ASPs	ASTs	ASQs	ASQuis	ASSs	equal mixture of 1–5	D3Gs	discordant mixture

For each dataset, 100 pedigrees were simulated using SLINK [51–53] for the genotype data at the disease locus and the SLINK utility program SUP [51, 54] for the marker genotypes.

Disease model $\{f_0, f_1, f_2\} = \{0.01, 0.1, 0.2\}$; p = 0.05.

Disease locus halfway between marker No. 50 and 51.

We used the following analysis options: 'imprinting on', 'algebraic calculation on/off', 'dimensions 5', 'saved models 0/5,000', 'number of replicates 1,000', 'maximization dense', 'penetrance restriction off', 'allfreq restriction off', 'analysis LOD', 'modcalc single', and 'calculate p value'.

families of the pedigree, to the disease locus. For the final remaining nuclear family of the pedigree or if the pedigree consists of only a single nuclear family, e.g. an ASP, a brute force calculation is employed. This calculation is done numerically and separately for each inheritance vector and for each assumed set of trait-model parameters. The LOD score of the currently analyzed family is stored, and the calculation continues with the next pedigree in the dataset. With the GHM software, many disease models are evaluated in a single program run during MOD-score analysis by repeating this step of the likelihood calculation. Our new algebraic procedure for calculating the disease-locus likelihood completely replaces the peeling algorithm, and it is applicable without additional modifications in case of inbreeding and marriage loops. It therefore significantly decreases the run time of a linkage analysis for any type of pedigree.

Maximization Options of GHM

The maximization routine of GHM first evaluates a set of predefined models. The user can choose between predefined grids with different densities. Moreover, the maximization can either be performed separately for each tested locus ('modcalc single' option) or jointly for the entire genetic region ('modcalc global' option). With modcalc single, calculation time can be saved by storing the trait-model-specific arrays of the disease-locus likelihood, which are needed for every considered genetic position. This option ('saved models') is especially useful when simulations are performed to obtain p values, which is already available with the previous version of GHM ('calculate p value' option [10]).

Simulations

To demonstrate the performance of our new method, we simulated datasets and compared the analysis run times of the algebraic algorithm to those of the peeling algorithm, which is employed by the original version that performs numeric calculation. Datasets either consisted of a single pedigree type, i.e. affected sibships with 2–6 siblings or three-generation pedigrees including unaffected pedigree members (discordant pedigrees), or mixtures of affected sibships. The speedup of the algebraic algorithm might be reduced by an increasing degree of discordance of the pedigrees, because this mostly leads to a larger number of inheritance vector classes as compared to their concordant counterparts (table 2). Therefore, we additionally considered an equal mixture of 4 discordant pedigree types: (a) discordant sib pairs, (b) discordant sib quadruplets, (c) discordant marriage loops (DML), and (d) discordant three-generation pedigrees (D3G). An overview of the simulated scenarios is given in table 3. Figure 3 depicts the pedigrees used for the discordant scenario including the one used in the D3G scenario (fig. 3d). Storing of arrays of the disease-locus likelihood, as already possible with the original GHM version (saved models option as mentioned above), was performed with the original algorithm (classic calculation mode). This was done to ensure a fair comparison to the classic calculation mode that makes use of run time-saving optimizations already implemented in the original GHM version. The saved models option was set to zero (no models saved) when using the algebraic algorithm (algebraic calculation mode), because it does not necessarily benefit from this option. It is of note that both our new

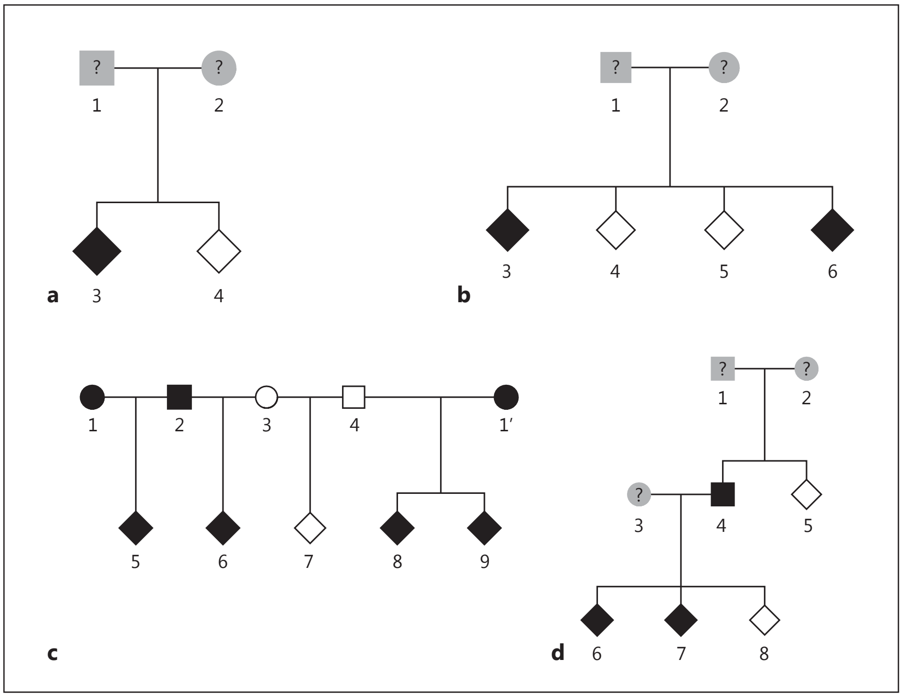


Fig. 3. Discordant pedigrees used in the simulations for run-time assessment. **a** Discordant sib pair; **b** discordant sib quadruplet; **c** discordant marriage loop; **d** discordant three-generation pedigree.

method and the saved models option need additional main memory. In case of the new method, this memory amount crucially depends on the size and phenotypic structure of the pedigrees, i.e. the number of inheritance vector classes across the whole dataset, whereas for the saved models option it depends on the number and size of the pedigrees. The peeling algorithm without the saved models option needs less memory albeit performing more floating point operations; it can still be used in case of insufficient main memory. Here we used a dense grid of disease models (option 'maximization dense'), because our new method should be especially useful when many disease models are evaluated, i.e. with a thorough maximization, which is likely to increase the power to map the disease gene under a complex mode of inheritance. In addition to the above-mentioned MOD-score analysis, p values were calculated (with the calculate p value option of GHM) by simulating 1,000 replicates generated under the null hypothesis of no linkage.

Run-Time Assessment

Run time was measured with the performance analysis tool *gprof* [46]. *gprof* measures the total amount of time spent executing each function of the program. Time due to system calls and waiting for CPU or I/O is not considered. Therefore, we additionally assessed the wall-clock time (WCT), which is the elapsed real time, i.e. the actual time taken from the start of the program run until the end. Because the WCT is obtained without any profiling steps, the program was run without any debugging options turned on. The speedup of our new method is obtained as follows:

$$\text{Speedup} = \frac{\text{run time with classic calculation mode}}{\text{run time with algebraic calculation mode}}.$$

All analyses were run on a single processor of the High Performance Computing – High Availability – Cluster (HPC-HA-Cluster) of the Helmholtz Zentrum München, equipped with IBM Intel Xeon X5690 6C, 3.46 GHz, 12 MB cache, 1,333 MHz 130 W processors in the compute nodes.

Results

The results of speedup due to the algebraic algorithm under the simulated scenarios for the analysis without calculating p values are shown in table 4, and those for the analysis with calculating p values are shown in table 5. Speedup is given based on run-time assessments measured by the performance analysis tool *gprof* as well as by measuring the WCT. Before looking at the speedups in detail, some technical aspects need to be considered prior to the interpretation of the results. In general, the *gprof* results reflect the speedup achieved by less time spent in the source code, which equals the number of instructions executed, but they do not include the time spent waiting for CPU and memory. Concerning GHM, the percentage of run time due to time waiting for CPU and memory increases with a larger number of scoring-function arrays saved in memory (saved models option) in case of the classic calculation mode, or with a larger number of inheritance vectors that must be considered when identify-

Table 4. Results of the run-time assessment without calculating p values, averaged over 3 program runs

	Run time, s				Speedup	
	WCT		*gprof*		WCT	*gprof*
	classic	algebraic	classic	algebraic		
Pedigree type						
ASPs	34.65	17.90	28.84	8.73	1.94	3.30
ASTs	95.75	23.76	73.97	15.13	4.03	4.89
ASQs	309.10	43.67	281.38	31.61	7.08	8.90
ASQuis	884.86	99.33	871.10	91.02	8.91	9.57
ASSs	4,523.33	392.67	3,372.38	378.24	11.52	8.92
Affected mixture	1,010.96	123.47	911.68	106.01	8.19	8.60
D3Gs	400.30	83.00	277.92	71.66	4.83	3.88
Discordant mixture	780.99	105.24	758.98	94.09	7.42	8.07

Classic = MOD-score analysis using the original GHM version; algebraic = MOD-score analysis using our new algebraic algorithm; *gprof* = execution time as measured by the profiling software *gprof*; ASQs = affected sib quadruplets; ASQuis = affected sib quintets; Affected mixture = mixture of 20 ASPs, ASTs, ASQs, ASQuis, and ASSs each; D3Gs = sample depicted in figure 3d; Discordant mixture = mixture of discordant pedigrees, 25 of each sort depicted in figure 3.

Table 5. Results of the run-time with calculating p values, averaged over 3 program runs

	Run time, h				Speedup	
	WCT		*gprof*		WCT	*gprof*
	classic	algebraic	classic	algebraic		
Pedigree type						
ASPs	9.28	5.49	5.18	2.22	1.69	2.33
ASTs	18.74	6.92	9.71	3.22	2.71	3.02
ASQs	62.85	11.68	17.96	4.95	5.38	3.63
ASQuis	243.48	31.85	26.98	9.01	7.64	2.99
ASSs	1,055.51	101.92	34.49	14.87	10.36	2.32
Affected mixture	278.97	33.98	28.81	9.56	8.21	3.01
D3Gs	177.80	20.39	36.84	8.29	8.72	4.44
Discordant mixture	294.83	29.36	29.20	10.69	10.04	2.73

See legend of table 4 for explanations.

ing inheritance vector classes in case of the algebraic calculation mode. For the latter, this is due to an increasing number of CPU memory cache misses caused by many crisscross copying processes of disease-locus-likelihood contributions of inheritance vectors of a given class into the corresponding memory cells of the scoring-function array. This copying process to complete the scoring-function has to be done for each inheritance vector, because $P_{complete}$, which will be multiplied with the scoring function, can be different for inheritance vectors of the same class. Hence, a larger number of inheritance vectors leads to more such copying processes, irrespective of the reduction factors as calculated in tables 1 and 2. When p values are calculated, this effect becomes more pronounced, as scoring-function arrays must be filled in this manner for every simulated replicate. In addition, it is of note that the results for the analyses without calculating p values are subject to a larger variance than those with calculating p values, because the analyses without calculating p values took only seconds to a few minutes to complete. With regard to the results in table 4 for the analyses without calculating p values, time waiting for CPU and memory was almost negligible. This is due to the fact that, in addition to time spent for the initial preparation of the dataset, time was predominantly spent for the initial identification of inheritance vector classes in case of the algebraic calculation mode or the initial numeric calculation of scoring-function arrays used for model saving in case of the classic calculation mode with the saved models option. Hence, the *gprof* speedups of the scenarios without calculating p values in table 4 were similar to their corresponding speedups calculated from the WCT. On the contrary, the *gprof* speedups of the scenarios with calculating p values in table 5 were quite constant over varying pedigree types due to a larger percentage of function calls invoked by the calculate p value option, which remained unchanged in the new GHM version. In addition, most of the computing time as measured by the WCT was spent waiting for CPU and memory (see explanation above). As the WCT is more relevant for users, since it is the actual time they have to wait for results, we concentrate our discussion of speedup on the WCT. As can be seen in table 4, the speedup for the analysis without calculating p values ranged from 1.94 for ASPs to 11.52 for affected sib sextets (ASSs). These speedups turned out to be roughly proportional to the reduction factors as calculated in table 1. The speedup for the mixture of nuclear families (8.19) was approximately the average of the individual speedups for each pedigree type. The speedups of the D3G and the discordant scenarios were 4.83 and 7.42, respectively, which are higher than would have been expected from the reduction factors in table 2. The fact that the increased computational effort of the peeling algorithm to calculate the disease-locus likelihoods of the D3G and DML pedigrees is avoided with our new algorithmic approach might be responsible for that. When p values were calculated, the speedups for the scenarios of nuclear families ranged from 1.69 for ASPs to 10.36 for ASSs (table 5), as was expected from the reduction factors calculated in table 1. Even though the classic calculation mode took advantage of model saving, whose effect should be more pronounced when simulating replicates to calculate p values, the speedups from table 5 for nuclear families were similar to those from table 4. The speedup for the mixture of nuclear families was 8.21, which was again roughly the average of the individual speedups for each pedigree type. The speedups of the D3G and the discordant scenarios were 8.72 and 10.04, respectively. Here, the speedups were higher compared to the results without calculating p values given in table 4. This is due to the fact that the percentage of time needed for peeling of the D3G and DML pedigrees with the classic calculation mode is even more pronounced when p values are calculated, because time due to initial calculations, i.e. the identification of inheritance vector classes for the algebraic calculation mode and the initial calculation of scoring-function arrays for the classic calculation mode, was negligible.

Discussion

The calculation of the disease-locus likelihood in linkage analysis is a complex task, because data on the observed genetic markers are often incomplete. This leads to a large number of possible disease-locus genotypes that must be considered in the likelihood. MOD-score analysis is a promising route to the genetic dissection of complex traits in the context of family studies. Although time-consuming, the evaluation of many disease models during a MOD-score analysis is essential, because it is thus likely to increase the power to map genes that act under a complex mode of inheritance, compared to a simple parametric (LOD-score) or nonparametric (NPL-score) analysis.

Our algebraic algorithm is inspired by the identity of the allele-sharing-based nonparametric likelihood and the parametric likelihood in the test for linkage. It is based on the concept of inheritance vectors. These are collapsed into inheritance vector classes, which turn out to be the distinct allele-sharing classes in the nonparametric context. In the Appendix section, we theoretically derive the allele-sharing classes for the example of an AST when an imprinting model is considered. This tedious way of identifying allele-sharing classes could principally be done for any type of pedigree considering affected as well as unaffected pedigree members in order to construct an allele-sharing-based test for linkage (see also Strauch [22]). Due to the above-mentioned identity, however, it is straightforward to express the allele-sharing probabilities as functions of the trait-model parameters f_0, f_1, f_2, and p, and to perform a MOD-score analysis, i.e. the parametric equivalent of the nonparametric test. The algebraic algorithm can thus be considered as a unified approach of parametric and nonparametric linkage methods. Previous work has shown that the MOD-score approach can outperform other linkage methods in terms of power [9]. One of the reasons for this finding is the fact that the performance of LOD scores crucially depends on the specification of the correct trait model, which is generally unknown when analyzing complex traits. This problem is circumvented by the MOD score which, in contrast to the simple LOD score, is maximized not only over the recombination fraction but also over trait-model parameters. However, the calculation of the disease-locus likelihood has to be done anew for every tested set of trait-model parameters, and it is the most time-consuming step in a MOD-score analysis. As a further complication, MOD scores are inflated when compared to LOD scores, and simulations to calculate p values have to be performed. Both aspects, extensive model testing and simulations to calculate p values, pose a challenge in regard to computation time and memory demands.

In this paper, we have presented a new algebraic algorithm that considerably reduces the run time of a MOD-score analysis. By storing unique IVDLGCs in a database common to all pedigrees in a dataset, the number of floating point operations and the memory demand of our new method are kept minimal, and similarities of family trees in terms of disease-locus-likelihood contributions can be exploited across the whole dataset. This is possible because the disease locus is treated separately from the marker loci when using a linkage analysis program such as GHM [10–13] that is based on the Lander-Green algorithm [25]. The speedup of a linkage analysis with GHM due to the algebraic algorithm depends on the number of different pedigree types, the complexity of the pedigrees, which is expressed by the number of inheritance vectors and classes, the number of replicates used to calculate p values, and the number of models saved in memory (saved models option) when running GHM in the classic calculation mode. For datasets consisting of only a single type of nuclear families, the speedup increased with the number of affected siblings and reached a factor >10 for ASSs in our analyses (tables 4, 5). Even in the case of ASPs, we achieved speedups by a factor of more than 1.5 (tables 4, 5). When using an equal mixture of nuclear families with different numbers of affected offspring, the speedups turned out to be the approximate average of the speedups of the individual nuclear family scenarios (tables 4, 5). In the D3G and the discordant scenarios, i.e. those scenarios with a larger degree of complexity of the pedigrees and a higher computational burden due to peeling and loop breaking for the classic calculation mode of GHM, the speedups increased from the analysis without calculating p values to those with calculating p values from 4.83 to a factor >8 for the D3G scenario, and from 7.42 to a factor >10 for the discordant scenario. The results thus clearly show that our new algorithm can substantially reduce the run time of a MOD-score analysis with GHM.

In the past, linkage analysis proved to be a valuable tool for identifying regions of the genome that harbor variants responsible for both Mendelian and complex diseases [2]. However, sequencing a rather large genetic region represented by the linkage signal to determine the causal variant was not feasible at that time. Nowadays, employing next-generation sequencing techniques allows for the identification of rare causal variants of putative complex-disease genes by combining an initial step of linkage analysis followed by fine mapping with association analysis. A major advantage of linkage methods as compared to methods in association analysis is that information across families can be combined, such that evidence for a causal role of a locus can accumulate even if different variants segregate at that locus in different families, which is known as allelic heterogeneity [47]. However, locus heterogeneity and/or penetrance heterogeneity, i.e. several allelic variants exist at the same locus each with different penetrances, can reduce the power of linkage analysis to map the disease gene. This problem can be diminished using large pedigrees, which can each be more homogeneous with respect to genetic variation than unrelated individuals or a sample of many small pedigrees [48, 49]. Admittedly, the GENEHUNTER software was originally designed for the analysis of small to moderately sized pedigrees ($2n - f \leq 20$ with n non-founders and f founders in a pedigree). Such pedigrees are easier to collect for diseases characterized by late onset, low penetrance, and diagnostic uncertainty. They are also more likely to reflect the genetic etiology of the disease in the general population [35]. The loss of power due to the uncertainty in penetrance values at the disease locus can be reduced by a maximization of the disease-locus likelihood over the trait-model parameters f_0, f_1, f_2, and p as it is done in a MOD-score analysis. Further robustness can be obtained by performing an affecteds-only analysis through recoding unaffected individuals as having an unknown phenotype. If the penetrance is low, little information is lost by ignoring the phenotype of unaffected pedigree members. The power of an affecteds-only MOD-score analysis can hence be higher, because the MOD-score distribution has fewer degrees of freedom as compared to the MOD score in an analysis that uses the phenotype of unaffected pedigree members. Even if pedigrees show locus and/or penetrance heterogeneity, it is likely that modest evidence for linkage can indeed narrow down the genetic region harboring the disease gene and can hence be used as a filter to focus on a more detailed association analysis of the variants in the region. In addition, using large samples of small pedigrees allows for the identification of hitherto unidentified genetic variants as risk factors for complex diseases (see de Visser et al. [50] for an example with

ASPs). Therefore, while linkage analysis of rare variants segregating in large pedigrees has proven to be a powerful approach, the analysis of smaller pedigrees can also be a promising route to discover genetic loci responsible for complex traits by the use of whole-exome or whole-genome sequence data. Irrespective of the assumed underlying genetic architecture of a given collection of small pedigrees, e.g. a large number of small-effect common variants, a large number of large-effect rare variants, or a mixture of both, GHM is well suited for the analysis of such data.

Extensive model testing, simulations to calculate p values, and the consideration of many genetic markers in a MOD-score analysis are indispensable to successfully map complex-disease genes in the context of family studies. Our new algebraic algorithm paves the way to an exceedingly efficient MOD-score analysis, because the evaluation of many sets of trait-model parameters and simulations to calculate p values are now feasible within a reasonable amount of time. Assuming, for example, an average speedup of 6.84 calculated from table 5, a geneticist doing a linkage study with MOD scores including simulations to determine p values can obtain results within a day instead of waiting a whole week for the analysis to finish. This further pushes ahead the maximum size of pedigrees that can still be analyzed.

GENEHUNTER-MODSCORE is thus a promising tool to identify rare causal variants segregating within families using next-generation-sequencing data. The algebraic algorithm is implemented in a new version of GHM that can be obtained for free from the following website: www.helmholtz-muenchen.de/ige/service/software-download/index.html.

Acknowledgements

This work was supported by grants Str643/4-1 and Str643/6-1 of the Deutsche Forschungsgemeinschaft (German Research Foundation). We thank Clemens Baumbach for his help with the figures. In addition, we thank the reviewers for their thoughtful comments, which have helped to improve the paper.

Appendix

Calculation of Allele-Sharing Classes for an AST Taking Imprinting into Account (see also Knapp [23] for the Formulation without Imprinting)

We are interested in the IBD sharing probability distribution of an AST at a diallelic disease locus with susceptibility allele D, normal allele d, and allele frequencies $p = P(D)$, $q = P(d) = 1 - \mathrm{p}$. Taking the parental origin of the alleles into account, 5 IBD configurations can be distinguished. These IBD configurations are identical to the inheritance vector classes. Table 1A presents the Mendelian probability for each IBD configuration and a representative sharing among the 3 sibs. Let w^D_i (i = 0, 1, 2^{pat}, 2^{mat}, and 3) denote the probability of the i-th configuration at the disease locus. Further, let D_p and D_m denote the paternal and maternal genotype at the disease locus. Let AST be the event that all 3 sibs are affected, and let IBD_i be the event that the sibs have IBD configuration i at the disease locus. For $k, l, m, n, \in \{D, d\}$, let $c_i^{(k,\,l,\,m,\,n)}$ denote the probability of the joint occurrence of AST and IBD^D_i, given that the paternal and maternal genotypes are (k, l) and (m, n). We hence get

$$\begin{aligned} c_i^{(k,\,l,\,m,\,n)} &= P\,(AST \cap IBD^D_i | D_{pat} = (k, l), D_{mat} = (m, n)) \\ &= P(AST | IBD^D_i, D_{pat} = (k, l), D_{mat} = (m, n)) \\ &\cdot P(IBD^D_i | D_{pat} = (k, l), D_{mat} = (m, n)), \end{aligned}$$

where $P(IBD^D_i | D_{pat} = (k, l), D_{mat} = (m, n))$ reduces to the Mendelian probability of the i-th IBD configuration, i.e. $P(IBD^D_i)$.

With *first-bits* $\in \mathcal{G}$, $\mathcal{G} = \{00, 01, 10, 11\}$ denoting the first two bits of the inheritance vector, which correspond to the outcome of the two meioses leading to the first offspring, we obtain

$$\begin{aligned} &P(AST | IBD^D_i, D_{pat} = (k, l), D_{mat} = (m, n)) \\ &\quad = \Sigma_{\textit{first-bits}\, \in\, \mathcal{G}}\, P(AST, \textit{first-bits} | IBD^D_i, D_{pat} = (k, l), D_{mat} \\ &\quad = (m, n)) \\ &\quad = \Sigma_{\textit{first-bits}\, \in\, \mathcal{G}}\, P(AST | \textit{first-bits}, IBD^D_i, D_{pat} = (k, l), D_{mat} \\ &\quad = (m, n)) \\ &\quad \cdot P\,(\textit{first-bits} | IBD^D_i, D_{pat} = (k, l), D_{mat} = (m, n)), \end{aligned}$$

where $P\,(\textit{first-bits} | IBD^D_i, D_{pat} = (k, l), D_{mat} = (m, n)) = 1/4$ for all *first-bits* $\in \{00, 01, 10, 11\}$.

Thus, we can write for $c_i^{(k,\,l,\,m,\,n)}$

$$\begin{aligned} c_i^{(k,\,l,\,m,\,n)} &= 1/4\, P(IBD^D_i)\, \Sigma_{\textit{first-bits}\, \in \mathrm{G}}\, P(AST | \textit{first-bits}, \\ &IBD^D_i, D_{pat} = (k, l), D_{mat} = (m, n)). \end{aligned}$$

Then with $\mathcal{J} = \{D, d\}^4$, it follows

$$\begin{aligned} &\sum_{(k,l,m,n)\in\mathcal{J}} c_i^{(k,l,m,n)}\, P\left(D_{pat} = (k,l), D_{mat} = (m,n)\right) \\ &= P\left(AST | IBD_i^D\right) \cdot P\left(IBD_i^D\right) = P\left(AST \cap IBD_i^D\right) \end{aligned}$$

and further

$$\begin{aligned} w_i^D &= P\left(IBD_i | AST\right) \\ &= \frac{\sum_{(k,l,m,n)\in\mathcal{J}} c_i^{(k,l,m,n)} P\left(D_{pat} = (k,l), D_{mat} = (m,n)\right)}{P\left(AST\right)}. \end{aligned}$$

For the 5 inheritance vector classes in the context of ASTs we obtain:

$$\begin{aligned} c_3^{(k,\,l,\,m,\,n)} &= 1/64\, ({f^3}_{km} + {f^3}_{kn} + {f^3}_{lm} + {f^3}_{ln}) \\ c_{2,pat}^{(k,\,l,\,m,\,n)} &= 3/64\, (f_{km} f_{kn} (f_{km} + f_{kn}) + f_{lm} f_{ln} (f_{lm} + f_{ln})) \\ c_{2,mat}^{(k,\,l,\,m,\,n)} &= 3/64\, (f_{km} f_{lm} (f_{km} + f_{lm}) + f_{kn} f_{ln} (f_{kn} + f_{ln})) \\ c_1^{(k,\,l,\,m,\,n)} &= 3/64\, (f_{km} f_{ln} (f_{km} + f_{ln}) + f_{kn} f_{lm} (f_{kn} + f_{lm})) \\ c_0^{(k,\,l,\,m,\,n)} &= 3/32\, (f_{km} f_{kn} (f_{lm} + f_{ln}) + f_{lm} f_{ln} (f_{km} + f_{kn})) \end{aligned}$$

$$w_3^D = \frac{1}{16P(AST)}\left(p^2 f_2^3 + pqf_{1,\,pat}^3 + pqf_{1,\,mat}^3 + q^2 f_0^3\right)$$

$$w_{2,\,pat}^D = \frac{3}{16P(AST)}\begin{pmatrix} p^3 f_2^3 + p^2 q\left(f_{1,\,pat} f_{1,\,mat}^2 + f_2^2 f_{1,\,pat} + f_2 f_{1,\,pat}^2\right) \\ + pq^2\left(f_{1,\,mat}^3 + f_{1,\,mat}^2 f_0 + f_{1,\,mat} f_0^2\right) \end{pmatrix}$$

$$w_{2,\,mat}^D = \frac{3}{16P(AST)}\begin{pmatrix} p^3 f_2^3 + p^2 q\left(f_{1\ ,\,pat}^2 f_{1,\,mat} + f_2^2 f_{1,\,mat} + f_2 f_{1,\,mat}^2\right) \\ + pq^2\left(f_{1,\,pat}^3 + f_{1,\,pat}^2\ f_0 + f_{1,\,pat} f_0^2\right) \end{pmatrix}$$

$$w_1^D = \frac{3}{16P(AST)}\begin{pmatrix} \left(p^2 f_2^2 + pqf_{1,\,pat}^2 + pqf_{1,\,mat}^2 + q^2 f_0^2\right) \\ \cdot\left(p^2 f_2 + pqf_{1,\,pat} + pqf_{1,\,mat} + q^2 f_0\right) \end{pmatrix}$$

$$w_0^D = \frac{3}{8P(AST)}\begin{pmatrix} p^4 f_2^3 + p^3 qf_2\left(f_2 f_{1,\,pat} + f_{1,\,pat}^2 + f_2 f_{1,\,mat} + f_{1,\,mat}^2\right) \\ + p^2 q^2 \begin{pmatrix} f_{1,\,pat}^3 + f_{1,\,mat}^3 + f_2 f_{1,\,pat} f_{1,\,mat} + f_2 f_{1,\,pat} f_0 \\ + f_2 f_{1,\,mat} f_0 + f_{1,\,pat} f_{1,\,mat} f_0 \end{pmatrix} \\ + pq^3 f_0\left(f_{1,\,pat}^2 + f_{1,\,pat} f_0 + f_{1,\,mat}^2 + f_{1,\,mat} f_0 + q^4 f_0^3\right) \end{pmatrix}.$$

Table 1A. IBD configurations for three affected siblings A, B, and C (adapted from Knapp [23])

IBD configuration/ inheritance vector class *i*	Alleles shared IBD by			Mendelian probability
	AB	AC	BC	
3	2	2	2	1/16
2^{pat}	2	1^{pat}	1^{pat}	3/16
2^{mat}	2	1^{mat}	1^{mat}	3/16
1	2	0	0	3/16
0	1^{pat}	0	1^{mat}	3/8

For each IBD configuration, i.e. inheritance vector class, *i*, the Mendelian probability and a representative sharing among the 3 siblings are given. Note that the 3 siblings A, B, and C cannot be distinguished, such that e.g. siblings A and C could be flipped, which reduces the number of inheritance vector classes. Hence, with 16 inheritance vectors for an AST, the Mendelian probability of e.g. inheritance vector class *i* = 1 is 3/16, because the sharing of 2 alleles IBD can take place either between A and B, A and C, or B and C, which does not have to be distinguished.

Table 2A. Mating types and conditional probabilities ci (adapted from Knapp [23])

No.	Parental mating type (pat × mat)	Probability of mating type	c_3	$c_{2,\text{ pat}}$	$c_{2,\text{ mat}}$	c_1	c_0	$\Sigma_i c_i$
1	DD × DD	p^4	$1/16\, f^3{}_2$	$3/16\, f^3{}_2$	$3/16\, f^3{}_2$	$3/16\, f^3{}_2$	$3/8\, f^3{}_2$	$f^3{}_2$
2	DD × Dd	$2p^3q$	$1/32\ (f^3{}_2 + f^3{}_{1,\text{ pat}})$	$3/32\ (f^2{}_2 f_{1,\text{ pat}} + f_2 f^2{}_{1,\text{ pat}})$	$3/32\ (f^3{}_2 + f^3{}_{1,\text{ pat}})$	$3/32\ (f^2{}_2 f_{1,\text{ pat}} + f_2 f^2{}_{1,\text{ pat}})$	$3/16\ (f^2{}_2 f_{1,\text{ pat}} + f_2 f^2{}_{1,\text{ pat}})$	$1/8\ (f_2 + f_{1,\text{ pat}})^3$
3	Dd × DD	$2p^3q$	$1/32\ (f^3{}_2 + f^3{}_{1,\text{ mat}})$	$3/32\ (f^3{}_2 + f^3{}_{1,\text{ mat}})$	$3/32\ (f^2{}_2 f_{1,\text{ mat}} + f_2 f^2{}_{1,\text{ mat}})$	$3/32\ (f^2{}_2 f_{1,\text{ mat}} + f_2 f^2{}_{1,\text{ mat}})$	$3/16\ (f^2{}_2 f_{1,\text{ mat}} + f_2 f^2{}_{1,\text{ mat}})$	$1/8\ (f_2 + f_{1,\text{ mat}})^3$
4	DD × dd	p^2q^2	$1/16\, f^3{}_{1,\text{ pat}}$	$3/16\, f^3{}_{1,\text{ pat}}$	$3/16\, f^3{}_{1,\text{ pat}}$	$3/16\, f^3{}_{1,\text{ pat}}$	$3/8\, f^3{}_{1,\text{ pat}}$	$f^3{}_{1,\text{ pat}}$
5	dd × DD	p^2q^2	$1/16\, f^3{}_{1,\text{ mat}}$	$3/16\, f^3{}_{1,\text{ mat}}$	$3/16\, f^3{}_{1,\text{ mat}}$	$3/16\, f^3{}_{1,\text{ mat}}$	$3/8\, f^3{}_{1,\text{ mat}}$	$f^3{}_{1,\text{ mat}}$
6	Dd × Dd	$4p^2q^2$	$1/64\ (f^3{}_2 + f^3{}_{1,\text{ pat}} + f^3{}_{1,\text{ mat}} + f^3{}_0)$	$3/64\ (f^2{}_2 f_{1,\text{ pat}} + f_2 f^2{}_{1,\text{ pat}} + f^2{}_{1,\text{ mat}} f_0 + f_{1,\text{ mat}} f^2{}_0)$	$3/64\ (f^2{}_2 f_{1,\text{ mat}} + f_2 f^2{}_{1,\text{ mat}} + f^2{}_{1,\text{ pat}} f_0 + f_{1,\text{ pat}} f^2{}_0)$	$3/64\ (f^2{}_2 f_0 + f_2 f^2{}_0 + f^2{}_{1,\text{ pat}} f_{1,\text{ mat}} + f_{1,\text{ pat}} f^2{}_{1,\text{ mat}})$	$3/32\ (f_2 f_{1,\text{ pat}} f_{1,\text{ mat}} + f_2 f_{1,\text{ pat}} f_0 + f_2 f_{1,\text{ mat}} f_0, + f_{1,\text{ pat}} f_{1,\text{ mat}} f_0)$	$1/64\ (f_2 + f_{1,\text{ pat}} + f_{1,\text{ mat}} + f_0)^3$
7	Dd × dd	$2pq^3$	$1/32\ (f^3{}_{1,\text{ pat}} + f^3{}_0)$	$3/32\ (f^3{}_{1,\text{ pat}} + f^3{}_0)$	$3/32\ (f^2{}_{1,\text{ pat}} f_0 + f_{1,\text{ pat}} f^2{}_0)$	$3/32\ (f^2{}_{1,\text{ pat}} f_0 + f_{1,\text{ pat}} f^2{}_0)$	$3/16\ (f^2{}_{1,\text{ pat}} f_0 + f_{1,\text{ pat}} f^2{}_0)$	$1/8\ (f_{1,\text{ pat}} + f_0)^3$
8	dd × Dd	$2pq^3$	$1/32\ (f^3{}_{1,\text{ mat}} + f^3{}_0)$	$3/32\ (f^2{}_{1,\text{ mat}} f_0 + f_{1,\text{ mat}} f^2{}_0)$	$3/32\ (f^3{}_{1,\text{ mat}} + f^3{}_0)$	$3/32\ (f^2{}_{1,\text{ mat}} f_0 + f_{1,\text{ mat}} f^2{}_0)$	$3/16\ (f^2{}_{1,\text{ mat}} f_0 + f_{1,\text{ mat}} f^2{}_0)$	$1/8\ (f_{1,\text{ mat}} + f_0)^3$
9	dd × dd	q^4	$1/16\, f^3{}_0$	$3/16\, f^3{}_0$	$3/16\, f^3{}_0$	$1/16\, f^3{}_0$	$3/8\, f^3{}_0$	$f^3{}_0$

A diallelic disease locus with susceptibility allele *D*, normal allele *d*, and allele frequencies $p = P(D)$, $q = P(d) = 1 - p$ is assumed. If the order of alleles within a parent is ignored, 9 mating types $(k, l, m, n) \in J$, with $J = \{D, d\}^4$ have to be distinguished. The mating type probabilities are given under the assumption of Hardy-Weinberg equilibrium at the disease locus. $c_i^{(k,\, l,\, m,\, n)}$ denotes the probability of the joint occurrence of 3 affected sibs that have IBD configuration *i* at the disease locus, given that the paternal and maternal genotypes are (k, l) and (m, n). $(f_0, f_{1,\text{ pat}}, f_{1,\text{ mat}},$ and $f_2)$ are the penetrances with f_i denoting the probability that an individual with *i* copies of the disease allele develops the disease. For the heterozygous individuals, separate penetrances for paternal and maternal transmission of the disease allele are distinguished to take imprinting into account.

References

1 Mohr J: A Study of Linkage in Man. Copenhagen, Munksgaards Forlag, 1954.

2 Bailey-Wilson JE, Wilson AF: Linkage analysis in the next-generation sequencing era. Hum Hered 2011;72:228–236.

3 Bowden DW: Will family studies return to prominence in human genetics and genomics? Rare variants and linkage analysis of complex traits. Genes Genom 2011;33:1–8.

4 Wilson AF, Ziegler A: Lessons learned from Genetic Analysis Workshop 17: transitioning from genome-wide association studies to whole-genome statistical genetic analysis. Genet Epidemiol 2011;35(suppl 1):S107–S114.

5 Morton NE: Sequential tests for the detection of linkage. Am J Hum Genet 1955;7:277–318.

6 Risch N: Segregation analysis incorporating linkage markers. I. Single-locus models with an application to type I diabetes. Am J Hum Genet 1984;36:363–386.

7 Clerget-Darpoux F, Bonaïti-Pellié C, Hochez J: Effects of misspecifying genetic parameters in lod score analysis. Biometrics 1986;42:393–399.

8 Risch N, Giuffra L: Model misspecification and multipoint linkage analysis. Hum Hered 1992;42:77–92.

9 Flaquer A, Strauch K: A comparison of different linkage statistics in small to moderate sized pedigrees with complex diseases. BMC Res Notes 2012;5:411.

10 Mattheisen M, Dietter J, Knapp M, Baur MP, Strauch K: Inferential testing for linkage with GENEHUNTER-MODSCORE: the impact of the pedigree structure on the null distribution of multipoint MOD scores. Genet Epidemiol 2008;32:73–83.

11 Dietter J, Mattheisen M, Fürst R, Rüschendorf F, Wienker TF, Strauch K: Linkage analysis using sex-specific recombination fractions with GENEHUNTER-MODSCORE. Bioinformatics 2007;23:64–70.

12 Strauch K, Fürst R, Rüschendorf F, Windemuth C, Dietter J, Flaquer A, Baur MP, Wienker TF: Linkage analysis of alcohol dependence using MOD scores. BMC Genet 2005;6(suppl 1):S162.

13 Strauch K: Parametric linkage analysis with automatic optimization of the disease model parameters. Am J Hum Genet 2003;73(suppl 1):A2624.

14 Flaquer A, Baumbach C, Piñero E, García Algas F, de la Fuente Sanchez MA, Rosell J, Toquero J, Alonso-Pulpon L, Garcia-Pavia P, Strauch K, Heine-Suñer D: Genome-wide linkage analysis of congenital heart defects using MOD score analysis identifies two novel loci. BMC Genet 2013;14:44.

15 Kruse LV, Nyegaard M, Christensen U, Møller-Larsen S, Haagerup A, Deleuran M, Hansen LG, Venø SK, Goossens D, Del-Favero J, Børglum AD: A genome-wide search for linkage to allergic rhinitis in Danish sib-pair families. Eur J Hum Genet 2012;20:965–972.

16 Christensen U, Møller-Larsen S, Nyegaard M, Haagerup A, Hedemand A, Brasch-Andersen C, Kruse TA, Corydon TJ, Deleuran M, Børglum AD: Linkage of atopic dermatitis to chromosomes 4q22, 3p24 and 3q21. Hum Genet 2009;126:549–557.

17 Schumacher J, Kaneva R, Jamra RA, et al: Genomewide scan and fine-mapping linkage studies in four European samples with bipolar affective disorder suggest a new susceptibility locus on chromosome 1p35–p36 and provides further evidence of loci on chromosome 4q31 and 6q24. Am J Hum Genet 2005;77:1102–1111.

18 Kurz T, Altmueller J, Strauch K, Rüschendorf F, Heinzmann A, Moffatt MF, Cookson WOCM, Inacio F, Nürnberg P, Stassen HH, Deichmann KA: A genome-wide screen on the genetics of atopy in a multiethnic European population reveals a major atopy locus on chromosome 3q21.3. Allergy 2005;60:192–199.

19 Knapp M, Seuchter SA, Baur MP: Linkage analysis in nuclear families. 2: Relationship between affected sib-pair tests and lod score analysis. Hum Hered 1994;44:44–51.

20 Holmans P: Asymptotic properties of affected-sib-pair linkage analysis. Am J Hum Genet 1993;52:362–374.

21 Suarez BK, Rice J, Reich T: The generalized sib pair IBD distribution: its use in the detection of linkage. Ann Hum Genet 1978;42:87–94.

22 Strauch K: MOD-score analysis with simple pedigrees: an overview of likelihood-based linkage methods. Hum Hered 2007;64:192–202.

23 Knapp M: A note on linkage analysis with affected sib triplets. Hum Hered 2005;59:21–25.

24 Elston RC, Stewart J: A general model for the genetic analysis of pedigree data. Hum Hered 1971;21:523–542.

25 Lander ES, Green P: Construction of multilocus genetic linkage maps in humans. Proc Natl Acad Sci USA 1987;84:2363–2367.

26 Lathrop GM, Lalouel JM, White RL: Construction of human linkage maps: likelihood calculations for multilocus linkage analysis. Genet Epidemiol 1986;3:39–52.

27 Lathrop GM, Lalouel JM, Julier C, Ott J: Strategies for multilocus linkage analysis in humans. Proc Natl Acad Sci USA 1984;81:3443–3446.

28 Lathrop GM, Lalouel JM: Easy calculations of lod scores and genetic risks on small computers. Am J Hum Genet 1984;36:460–465.

29 Schäffer AA, Gupta SK, Shriram K, Cottingham RW Jr: Avoiding recomputation in linkage analysis. Hum Hered 1994;44:225–237.

30 Cottingham RW Jr, Idury RM, Schäffer AA: Faster sequential genetic linkage computations. Am J Hum Genet 1993;53:252–263.

31 O'Connell JR: Rapid multipoint linkage analysis via inheritance vectors in the Elston-Stewart algorithm. Hum Hered 2001;51:226–240.

32 O'Connell JR, Weeks DE: The VITESSE algorithm for rapid exact multilocus linkage analysis via genotype set-recoding and fuzzy inheritance. Nat Genet 1995;11:402–408.

33 Gertz EM, Hiekkalinna T, Digabel SL, Audet C, Terwilliger JD, Schäffer AA: PSEUDOMARKER 2.0: efficient computation of likelihoods using NOMAD. BMC Bioinformatics 2014;15:47.

34 Hiekkalinna T, Schäffer AA, Lambert B, Norrgrann P, Göring HH, Terwilliger JD: PSEUDOMARKER: a powerful program for joint linkage and/or linkage disequilibrium analysis on mixtures of singletons and related individuals. Hum Hered 2011;71:256–266.

35 Kruglyak L, Daly MJ, Reeve-Daly MP, Lander ES: Parametric and nonparametric linkage analysis: a unified multipoint approach. Am J Hum Genet 1996;58:1347–1363.

36 Gudbjartsson DF, Thorvaldsson T, Kong A, Gunnarsson G, Ingolfsdottir A: Allegro version 2. Nat Genet. 2005;37:1015–1016.

37 Gudbjartsson DF, Jonasson K, Frigge ML, Kong A: Allegro, a new computer program for multipoint linkage analysis. Nat Genet 2000;25:12–13.

38 Abecasis GR, Cherny SS, Cookson WO, Cardon LR: Merlin – rapid analysis of dense genetic maps using sparse gene flow trees. Nat Genet 2002;30:97–101.

39 Markianos K, Daly MJ, Kruglyak L: Efficient multipoint linkage analysis through reduction of inheritance space. Am J Hum Genet 2001;68:963–977.

40 Idury RM, Elston RC: A faster and more general hidden Markov model algorithm for multipoint likelihood calculations. Hum Hered 1997;47:197–202.

41 Kruglyak L, Lander ES: Faster multipoint linkage analysis using Fourier transforms. J Comput Biol 1998;5:1–7.

42 Nyholt DR: GENEHUNTER: your 'one-stop shop' for statistical genetic analysis? Hum Hered 2002;53:2–7.

43 Whittemore AS, Halpern J: A class of tests for linkage using affected pedigree members. Biometrics 1994;50:118–127.

44 McPeek MS: Optimal allele-sharing statistics for genetic mapping using affected relatives. Genet Epidemiol 1999;16:225–249.

45 Strauch K, Fimmers R, Kurz T, Deichmann KA, Wienker TF, Baur MP: Parametric and nonparametric multipoint linkage analysis with imprinting and two-locus-trait models: application to mite sensitization. Am J Hum Genet 2000;66:1945–1957.

46 Graham SL, Kessler PB, McKusick MK: An execution profiler for modular programs. Software Pract Exper 1983;13:671–685.

47 Balding DJ: A tutorial on statistical methods for population association studies. Nat Rev Genet 2006;7:781–791.

48 Wijsman EM: The role of large pedigrees in an era of high-throughput sequencing. Hum Genet 2012;131:1555–1563.

49 Terwilliger JD: On the resolution and feasibility of genome scanning approaches. Adv Genet 2001;42:351–391.

50 de Visser MCH, van Minkelen R, van Marion V, den Heijer M, Eikenboom J, Vos HL, Slagboom PE, Houwing-Duistermaat JJ, Rosendaal FR, Bertina RM: Genome-wide linkage scan in affected sibling pairs identifies novel susceptibility region for venous thromboembolism: Genetics In Familial Thrombosis study. J Thromb Haemost 2013;11:1474–1484.

51 Schäffer AA, Lemire M, Ott J, Lathrop GM, Weeks DE: Coordinated conditional simulation with SLINK and SUP of many markers linked or associated to a trait in large pedigrees. Hum Hered 2011;71:126–134.

52 Weeks DE, Lehner T, Squires-Wheeler E, Kaufmann C, Ott J: Measuring the inflation of the lod score due to its maximization over model parameter values in human linkage analysis. Genet Epidemiol 1990;7:237–243.

53 Ott J: Computer-simulation methods in human linkage analysis. Proc Natl Acad Sci USA 1989;86:4175–4178.

54 Lemire M: SUP: an extension to SLINK to allow a larger number of marker loci to be simulated in pedigrees conditional on trait values. BMC Genet 2006;7:40.