University of Oxford

Incremental Pedigree Likelihood Calculation

by

Shuo Zhang

Exeter College

A dissertation submitted in partial fulfilment of the degree of Master of Science in Applied Statistics.

Department of Statistics, 1 South Parks Road,
Oxford OX1 3TG

September 2007

This is my own work (except where otherwise indicated)

Candidate: Shuo Zhang

Signed:................................

Date:..................................

Abstract

We consider the problems in getting the probability of the data based on a particular pedigree use the DNA marker information. The ”probability” here is inferred more as the ”likelihood” since it gives a measurement in getting such a pedigree based on the data. We implement the two widely used algorithm in linkage analysis: the Elston-Stewart algorithm and the Lander-Green algorithm. We use the significant results from the graphical models and the Bayesian probability networks to investigate the pedigree. The graphical models can not only be used as a language but also as an important tool in analyzing the pedigree. We simulate a few pedigree and get their likelihood based on the two algorithms and through this procedure, we explore the features of each algorithm and potential improving methods. We propose a way in generating an unobserved person’s genotype according to his offspring and combined it with the Elston-Stewart algorithm as well as the Lander-Green algorithm. We also implement an incremental algorithm to construct the contrasts to a pedigree under research. This incremental algorithm will give us an easy approach to get the likelihood based on the new pedigree without re-calculating everything.

Acknowledgements

I would like to thank my supervisor, Professor Jotun Hein, for his support, guidance and comments during this half year. I would also like to thank Dr.Rune Lyngsoe for his guidance and advice. For help with technical knowledge, I would like to thank Susan Hutchinson, Mr Yuqiang Zhou. Finally, I would like to thank my family and my friend Alonso for their encouragement and support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Linkage analysis is a well established method for studying the relationship between the pattern of the occurrence of a given biological trait such as a disease and the inheritance pattern of certain genes in a given family. This time we concentrate on the inheritance pattern from generation to generation. Linkage analysis is performed on a pedigree, and the analysis of human pedigree relationship has generated many challenging computational problems. This is because in most genes mapping studies the DNA sequence of each individual is measured imperfectly: some are missing data; some are with great uncertainty, while the others are not observed indirectly such as our ancestors' profiles. In each situation, there may be a large amount of DNA sequence compatible with the observed data, so identifying the most likely DNA sequence configurations might require simultaneously consideration of many individuals, not only the individuals related recently and directly. The linkage analysis is based on multi point analysis which derived from the single point ones by taking the probability of recombination into account.

There are two things which interest us in linkage analysis besides the quality of data: one is the recombination fraction and the other is the likelihood based on this pedigree when given other things informative. The recombination rate is related to the genetic map distance between two loci, and there are several link functions between the genetic map distance and the recombination fraction, say, Poisson process with rate 1 per Morgan, the measurement of genetic map distance. There are other non-parametric ways in getting the recombination rate such as the Gaussian-Newton method and the EM algorithm etc and the EM algorithm may be the best way in getting the most likely recombination rate. Based on these existed methods in getting the recombination fraction, we can concentrate on the other thing that interests us: the likelihood.

There are many ways in investigating this problems within which the widely-used algorithms are proposed by Elston-Stewart and Lander-Green respectively. They were demonstrated in the 1970s and 1980s and later on a few enhancements were performed to make things better. With the development in the graphical models, it is highly probable for us to create more flexible framework allowing for efficient algorithm. The graphical models is not only a way in expressing pedigree visually but also it benefit us in implementing the incremental algorithm. We use the graphical models in peeling off pedigree and identifying newly insert figures. By this mean, we can accelerate our approach more quickly without repeating inputting the same thing.

The Elston-Stewart algorithm(1971) is the first general algorithm for rapid pedigree likelihood calculation. The basic assumption in this algorithm is that the phenotype of each individual only depends on its own associated genotypes. It and its variations work well for the oligogenic case, but is computationally difficult in some mixed model cases, which mix over all possible genotypes of individuals in each family unit. This algorithm is designed to deal with pedigree with a large number of members but a small number of loci. Many improvements to the basic algorithm have been proposed. For example Cannings *et al.* (1978) showed how the method could be applied to complex pedigrees, even with inbreeding and loops. Lange and Boehnke (1983) showed that the likelihood could be updated one individual at a time, rather than a nuclear family and different sequences of updates could result in dramatically different computing time and memory requirements. In the 1990s, further enhancements to the Elston-Stewart algorithm were discovered.

Another algorithm suggested by Lander and Green(1987) is based on the Hidden Markov Models and the inheritance vectors. On the contrary to Elston-Stewart algorithm, it is better used in a pedigree consisting of fewer members but a large amount of loci. The Lander-Green algorithm supports non-parametric linkage analysis, and as refer to the missing data, it performs much better than Elston-Stewart algorithm. In 1974, Morton and McLean proposed a model which worked for both cases, but the validity of the assumption underlying the model is not easy to justify. Enhancements are done in two aspects, with one resulted from the observation that there are many redundancies within inheritance vector space so that inheritance vectors can be grouped to speed up calculation. This was done by Kruglyak *et al* in (1996) and Gudbjartsson *et al.* (2000) as well as other people such as Abecasis *et al.* (2002). The other approach focuses on the manipulation of transition matrices, used for the calculation of conditional inheritance vector distributions at neighboring locations. Two distinct approaches have proved very successfully at speeding up this step of the calculation: either divide-and-conquer algorithm by Indury and Elston in 1997 or Fast Fourier Transform by Krusglyak and Lander in 1998 can reduce the computational cost of generating these conditional distributions. Although current implementations of the Lander-Green algorithm can comfortably handle thousands of genetic loci, still it is not enough to SNP (Single-Nucleotide polymorphism) markers which requires millions of gene loci calculation.

Both Elston-Stewart algorithm and Lander-Green algorithm are implemented in some computer packages such as LINKAGE, VITESSE and GENEHUNTER. Since neither of the two algorithms can handle a large pedigree with many members and loci simultaneously, people began to search new ways leading to the large pedigree problem. An intuitive response is to combine the two algorithms thus to get a newly-born one but it is nearly impossible since they are investigating in different directions. The Monte-Carlo based methods have been employed successfully in linkage analysis, including Simulated Annealing, the Gibbs Sampler and Sequential Imputation. Although it works well for a large pedigree, refine to the algorithm is still needed and understanding of the properties is necessary.

To calculate likelihood ratios between similar pedigrees, methods need to be developed which do not traverse the entire pedigree but are based on the idea that the relevant effect of changes only affect small subsets of the pedigree. This can either be exactly or approximately true. The incremental algorithm is preferable in constructing the contrasting pedigree based on an existing pedigree. Strictly speaking this is not a new algorithm but we would like to call it as an 'accessories' to the existing algorithms. This is an idea based on saving the old information triple by triple instead of storing them as a multiplication of some family terms and with properties of graphical models, we are able to get a new likelihood easily without redundant work and thanks to the branches in graphical models we can make the computer smart enough to identify the updated information.

In this project we implement the Elston-Stewart algorithm and the Lander-Green algorithm in C programming language. In order to check the reliability, we also implement the exhaustive algorithm to give a check. Then we proposed an incremental algorithm. This algorithm is used to construct contrasting pedigree to the pedigree which is under research. With this algorithm we can get the likelihood for the new contrasting pedigree without re-calculating all the triples or markers we have. In Chapter 2 we introduce basic concepts from genetics; in Chapter 3 we illustrate how to use graphic models in the pedigree, including using it to represent the pedigree, process of meiosis and inheritance pattern. It makes the complicated pedigree understandable. In Chapter 4 we focus on the three existed algorithm and the incremental algorithm. We also give some simulations in explaining them. Finally, in Chapter 5, we give a conclusion based on our work.

Chapter 2

Mendelian Models

2.1 Some Basic Genetics

A chromosome is a single large macromolecule of DNA, which constitutes a physically organized form of DNA in a cell. It is a very long, continuous piece of DNA (a single DNA molecule). In diploid individuals such as mammals, DNA in each normal cell is packed into pairs of chromosomes. Human beings, for example have 23 pairs and 22 of them are called the autosomes, with the remaining pair are the sex chromosomes. For a given individual, one chromosome in each pair derives from the DNA of his mother and the other from the DNA of his father. A specific segment of chromosome is known as a locus, and we typically refer to the individual's DNA at this locus as his gene. Different variants of a gene are called alleles. Then unordered pair of alleles at any locus is known as the genotype and the potentially observable characteristic is the phenotype, for example eye color, blood type, affected/normal etc. The term ordered genotype, or full genotype, can be used to keep track of the parental origins of each allele in a genotype, but we don't distinguish them since we just have to sum all the possibilities.

If both alleles are of the same type, we say that the genotype is homozygous otherwise heterozygous. If a homozygote has the same phenotype as a heterozygous type with which it has only one allele in common, we say that the common allele dominates the other allele in the heterozygote. Alternatively, the allele that is not shared is said to be recessive to the common one. An easy example is human ABO blood group, simplified to a three-allele genetic system.

Genotype	AA	AO	AB	BB	BO	O
Phenotype	A	A	AB	B	B	O

Table 2.1: The six phenotypes for the human ABO system and the four corresponding expressed phenotypes

Table 2.1 illustrates the relationships between phenotypes and genotypes for human blood type.

The homozygous genotypes are AA, BB and OO, while the heterozygous types are AO, AB and BO. AA and AO have the same phenotypes, so A dominates O or O is recessive to A. Similarly we can see that O is also recessive to B. The genotype AB has its own phenotype, and we conclude that A and B are co-dominant.

2.1.1 Mendelian Inheritance

We express Mendel's first law as follows:

1. alleles account for variations in inherited characteristics;
2. for each characteristic, an organism inherits two alleles, one from each parent;
3. alleles could express the trait: dominant or recessive, however, there are incomplete dominance;
4. two alleles for each characteristic segregate during gamete production.

Furthermore, an individual passes of his two genes to each of his children, independently for different segregations and independently of segregations from the other parent. When genes segregate with equal probability 1/2, we have what is known as Mendelian segregation. Mendel's second law states that segregations of genes at different loci are independent. This is now known not to be true in general: these segregations may be correlated if the loci are close together on the same chromosome or linked.

2.2 Recombination

Sperm and egg end up with the DNA they get through a process called meiosis. When cells normally make new copies of themselves, each of the new cells ends up with 46 chromosomes for humans. Recombination events occur during meiosis. Breaks occur at several random positions which allow for the exchange of segments of chromosome within the pair. This is called crossing over and refers to the interchange of genetic material between the two homologous chromosomes.

Recombination is due to the independent assortment of chromosomes during meiosis. However, alleles that are on the same chromosome are more likely to be inherited together, and are said to be linked. Because there is some crossing over of DNA when the chromosomes segregate, alleles on the same chromosome can be separated and go to different offspring cells. There is a greater probability of this happening if the alleles are far apart on the chromosome, as it is more likely that a cross-over will occur between them.

The resulting chromosomes, which are mixtures of the maternal and paternal chromosome segments, separate and one of each pair is passed to the gamete-the genetic contribution form a single parent to the next generation, sperm or egg. We here induce the term haplotype to refer to a listing of all possible alleles in a single gamete at a given number of loci. They may lie on the same chromosome, but not necessarily. For example if we ignore the order, there are 3 haplotype for blood group, A, B and O. The correlation in segregation between liked loci is due to the fact that it is highly unlikely that a crossover will occur between two loci which are physically close on the chromosome. Loci which are 'far apart' (or on different chromosomes) are more likely to segregate independently in accordance with Mendel's second law. A straightforward example is shown in Figure 2.1.

The recombination fraction r between two loci is defined as the probability that the genes segregating to the gamete at these loci come from different parental chromosomes. For loci close together, r is nearly 0; which when loci are far away, r tends to be 1/2, indicating that the loci are segregating independently. That is to say, under assumptions of the meiosis model for most diploid species, r ranges from 0 to 1/2.

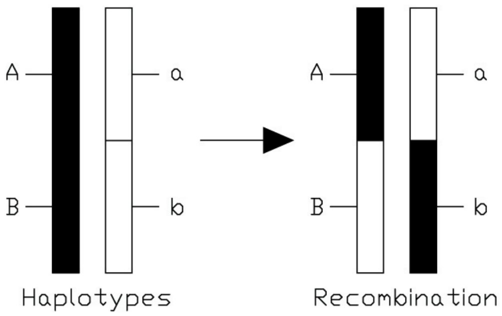

Figure 2.1: Schematic representation of meiosis showing the chromosomes which form the gametes containing some maternal and some paternal DNA after crossing over has occurred.

Let M_1, M_2,...M_n be n genetic loci, listed in the correct chromosomal order. Given information about r_i, $i = 1$ to $n - 1$ denote the recombination fraction between adjacent loci M_i and M_{i+1}. Thus we could get a vector $R = (r_1, r_2, ...r_{m-1})$ for different locus-pair.

Naturally it is reasonable and important for us to consider about the number of recombination in each chromosome. Furthermore, we know that the number of recombination is related to the probability of recombination and the probability of a recombination happening between neighboring loci will determine the total expected number of recombination. Thus if 200 loci were considered from a chromosome, the average probability of a recombination between neighboring loci would be 2.63/199, which is around 0.01. It would still be possible, though, to have a recombination happening between each pair of neighboring loci. However, though possible it would be highly improbable (probability of about 10^{-374}). So number of recombination shouldn't be controlled by bounds but merely by probabilities, which means it is theoretically possible that if we have 100 loci, 99 times of recombination would have happened.

2.3 Other Prerequisite Knowledge

In this part we introduce some related background knowledge under which we build our models. There are three definitions we would like to list here.

- **Hardy-Weinberg principle** It is a relationship between the frequencies of alleles and the genotype of a population. The occurrence of a genotype, perhaps one associated with a disease, stays constant unless matings are non-random or inappropriate, or mutations accumulate. Therefore, the frequency of genotypes and the frequency of alleles are said to be at 'genetic equilibrium'. Genetic equilibrium is a basic principle of population genetics.

- **Linkage equilibrium** It describes the situation in which the haplotype frequencies in a population have the same value that they would have if the genes at each locus were combined at random. In other words, it occurs when linkage disequilibrium is at zero. Linkage equilibrium can be thought of as the two locus version of the Hardy-Weinberg ratio, but it is a property of haplotype, not genotypes. A diploid individual has two haplotype, and at the equilibrium the genotypes at each locus will be in Hardy-Weinberg proportions while the haplotype are at linkage equilibrium.

- **Haldane's Model** It assumes independence of genes at different loci; if the selection intensity is 0.1 for each gene moving towards fixation, and there are N such genes, then the reproductive capacity of the species will be lowered to 0.9^N times the original capacity. Therefore, if it is necessary for the population to fix more than one gene, it may not have reproductive capacity to counter the deaths.

Chapter 3

Graphic Models

3.1 Graphical Definitions

We introduce fundamental concepts in pedigree and graphical models. We will define a pedigree or a genealogy to be a group of individuals together with a full specification of all the relationships along with the members among them (Thompson, 1986). The set of familial relationships among a group of individuals forms what is commonly known as a pedigree and a variety of graphical representations have been developed for handling pedigrees in a precise and consistent manner. We define a pair of pedigree members to be spouses only if they have mutual offspring in the pedigree and every such pairing is called a marriage. Those without parents are called the founders of the pedigree and these, by definition are unrelated. Pedigrees are commonly represented graphically, although not always strictly as a graph. A simple example is shown in Figure 3.1.

A graph, G=(V, E) is a structure consisting of a finite set V of vertices (or so called nodes) and a finite set E of edges between these vertices. The vertices representing the variables while the edges representing the links between these variables. Edges can be either directed, with arrows indicating the direction of the link or undirected. Directed edges are also called arcs and a graph with arcs is called a directed graph. This indicates that we can represent a pedigree by a directed graph (Lange and Elston, 1975) as shown in Figure 3.2. where the nodes denote individuals and the arcs connect individuals to their offspring. It is an obvious way to show a

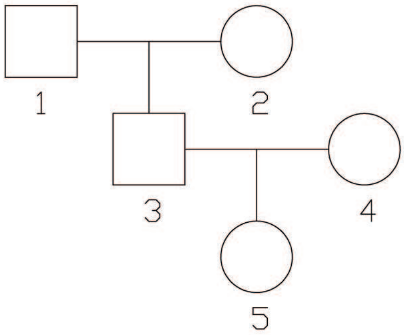

Figure 3.1: A standard graphical representation of a simple pedigree of 5 individuals. Females are represented by circles and males by squares. Individuals 1, 2 are the baseline founders while 3 is the recent founder. Individual 5 is the final and he has no marriage.

standard representation of the transmission of genes from parents to offspring.

Although for probability calculations on pedigrees the vertices represent variables for properties of individuals rather than the individuals themselves, the parent-offspring analogy used generally in graphical modeling, in effect, becomes literal for pedigree analysis. Figure 3.1 and Figure 3.2 are both marriage node graphs, which means the parents might not be the biological parents of their offspring. Figure 3.2 is also known as a relationship graph with nodes representing individuals and directed edges connecting individuals to their offspring.

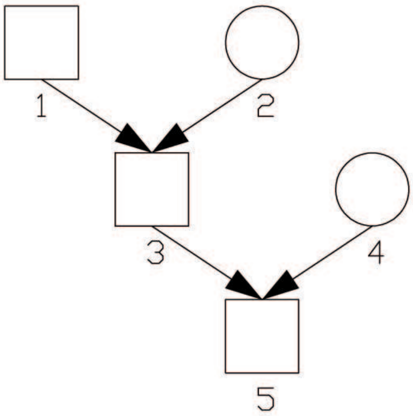

Figure 3.2: Pedigree of previous figure drawn as a directed graph with nodes representing individuals and directed edges connecting individuals to their offspring

3.2 Bayesian Networks

Realize that we could split the complicated problem into small manageable components; an immediate thinking is to treat a pedigree generation by generation, family by family. The advantage in graphic model is that a complex problem can be represented in a graphical form which can then inform the development of efficient computational algorithms for performing calculations. A better approach would be Bayesian network which might lead to essentially simplify the computations.

We have to be careful for our applications that the nodes represented in the pedigree graph may not be the child's biology parents. For instance, for 3 and 5, we say 3 is a parent of 5 since there is a directed edge from 3 to 5, or 5 is a child of 3. In contrast with the biological interpretation of these terms, a node in a graph can have more than two parents, for example adopted child. To avoid the ambiguity, we will further refer the term using expressions like graph or bio parent. In the later likelihood calculation, we will see that a graph but not bio parent would simplify the likelihood calculation and get a zero probability in forming such a pedigree.

Recall that a graph $G = (V, E)$ is a collection of nodes and edges which can be either directed (arcs) or undirected. A walk is a sequence of $k \geq 0$ nodes v_0,v_1,...,v_k that satisfies v_i connected with v_{i+1} for each $i = 0, .., n - 1$. A path is a walk without repeated nodes. If there is a path from a to b we say that a lead to b and write $a \mapsto b$.If all edges of the path are undirected, the path is undirected; and the path is directed if all edges are directed. If $a \mapsto b$ and $b \mapsto a$ doesn't hold, we say that a is an ancestor of b and b is a descendent of a. A walk beginning and ending with the same node is a cycle or loop. If all the edges of a graph are directed and if the graph has no directed cycles, it is a directed acyclic graph or DAG. A graph is complete if all nodes are joined by an arc or a line. A subset is complete if it induces a complete subgraph. A complete subset that is maximal is called a clique. A graph is connected if there is a walk between any pair of nodes. Unless otherwise stated, it will be assumed that all graphs are connected.

Returning to our graphical representation for a pedigree, we state the following requirements:

- The maximum cardinality for a clique is 2: compromised of a parent and a child, which require a moral graph without loop or inbreeding;
- The pedigree graph should be a DAG, with arcs connected ancestors and children; here we don't distinguish graphical parent from bio parent since we don't know the hidden truth.

A Bayesian network (or a belief network) is a probabilistic graphical model that represents a set of variables and their probabilistic independencies. For example, a Bayesian network can be used to calculate the probability of a patient having a specific disease, given the absence or presence of certain symptoms, if the probabilistic independencies between symptoms and disease as encoded by the graph hold. Here we use Bayesian network to calculate the pedigree likelihood based on the observed genotypes, knowing that there are some causal relationships between generations and they could be interpreted by graph. The term "Bayesian networks" emphasizes three aspects:

1. the often subjective nature of the input information;
2. the reliance on Bayes's conditioning as the basis for updating information;
3. the distinction between causal and evidential modes of reasoning.

Formally, Bayesian networks are DAG whose nodes represent variables, and whose arcs encode conditional independencies between the variables. Nodes can represent any kind of variable, be it a measured parameter, a latent variable or a hypothesis. They are not restricted to representing random variables, which forms the "Bayesian" aspect of a Bayesian network. Back to our pedigree graph, we only require a node set V, where the nodes represent random variables. The set of parent nodes of a node $X_i \in V$ is denoted by $pa(X_i)$. A directed acyclic graph is a Bayesian Network relative to a set of variables if the joint distribution of the node values can be written as the product of the local distributions of each node and its parents. A DAG is a Bayesian Network relative to a set of variables if the joint distribution of the node values can be written as the product of the local distributions of each node and its parents:

$$f(x) = \prod_{X_i \in V} f(X_i | X_{pa(X_i)})$$

If node X_i has no parents, its local probability distribution is said to be unconditional, otherwise it is conditional. So it then holds that any node, given the values at its parents, is conditionally independent of any other nodes. This is known as the directed local Markov property. It can be seen that using a Bayesian network we can specify the joint distribution completely from the associated DAG and the conditional distributions of each node given its parents. We give the nodes without ancestor probability 1 and later on assign a prior probability on them.

3.3 Bayesian Network Representations for Pedigrees

There are several ways in using the networks and each of them has different properties. Mostly used ones are genotype network, marriage network and segregation network. We describe the genotype network initially since the likelihood calculation base on each haplotype is the term that interests us the most.

3.3.1 The Genotype Networks

We choose this network to expand and research our pedigree based on the trait of our data and algorithms: since our algorithms are based on Mendelian model and all data are binary. Another point we have to figure out is that the genotype networks holds only under the Mendelian inheritance model. A typical genotype network is shown in Figure 3.3 .

Each node i in this DAG represents a random variable assigning a genotype to individual i. The graph parents here are considered also as the biological parents of the nonfounders. Genotype G_i for each individual i denotes the person's genotype at our interested locus, so that G$_i$ $=$ (L$_{i^1}$,L_{i^0}), where L_{i^1}, L_{i^0} assigning the allelic type in the gene inherited by individual i from his father and mother respectively. Also we could decompose the genotype as $G_i = (G_{m_i}, G_{p_i})$, where each component denotes an allele got from mother or father. We will use the genotype networks throughout the work and there will be an extra thing in the genotype networks shown later, the observed genotypes for some individuals in the pedigree.

Now we derive the probability for such a genotype network. First of all we need to get probabilities for founders genotypes as well as the transmission probabilities. If we denote the

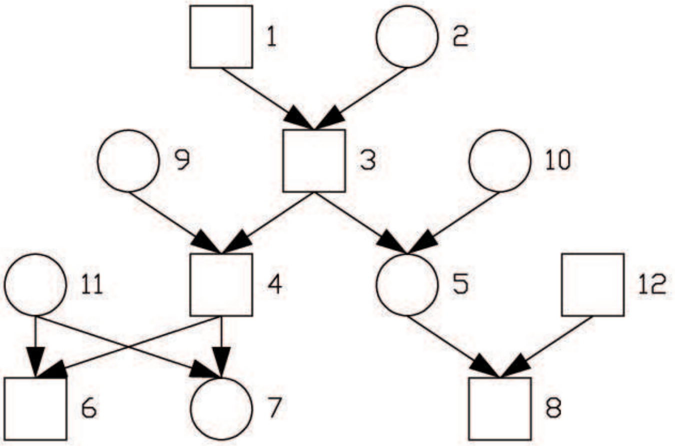

Figure 3.3: A genotype network

probabilities of the founder genotypes by π and the transmission probabilities by:

$$\tau(g_i|g_{m_i}, g_{p_i}) = P(G_i = g_i|G_{m_i} = g_{m_i}, G_{p_i} = g_{p_i})$$

So the likelihood based on the genotype network is:

$$P(g_1, ...g_m) = \prod_{i \in F} \pi(g_i) \prod_{j \notin F} \tau(g_i|g_{m_i}, g_{p_i}) \tag{3.1}$$

where F is the set of individuals which are founders of the pedigree and the transmission probabilities τ are under the Mendelian models.

3.3.2 Genetic Descent Graph

A descent graph for the locus of interest is a model which species which allele was inherited by each individual from each parent. It is a haplotype graph nd quite straightforward in each single locus however in multi point linkage analysis we hardly use it since it requires too many graphs to be taken into consideration. Combining the neighboring descent graph together we could inference whether there is a recombination happening or not. A genetic descent graph is shown in Figure 3.4.

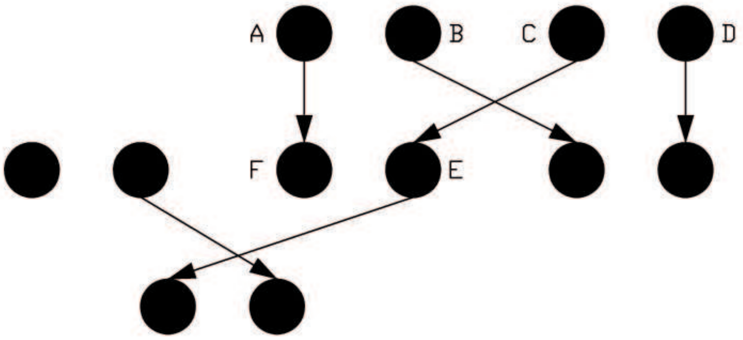

Figure 3.4: A genetic descent graph

A genetic descent graph give us a clear idea on the origins of an individual's alleles at each locus. We will see later in the Lander-Green algorithm we could use the genetic descent graph combined with the genotype networks to investigate the inheritance pattern in a pedigree.

Chapter 4

Algorithms in Calculating Pedigree Likelihood

4.1 Background

Linkage analysis is an integral part of investigations into the genetics of pedigree information, complex diseases and anthropology research. It has been used to successfully calculate the likelihood on a pedigree, such as DNA test and mapping many disease genes. We emphasize again that genetic linkage occurs when particular alleles are inherited jointly. In parametric linkage analysis a particular genetic model for the trait is assumed and the likelihood of the pedigree data is computed for the trait locus placed at various positions along a framework map. Multipoint likelihoods use all the available markers and take recombination into consideration.

Besides the raw materials of pedigree structure and observed phenotypes, a genetic model is a prerequisite for likelihood calculation. At its most elementary level, a model postulates the number of loci necessary to explain the phenotypes. For purposes of discussion, it is convenient to use the term ”genotypes” when discussing the multilocus, ordered genotypes of an underlying model. Because ordered genotypes preserve phase, they are preferable to unordered genotypes for theoretical and computational purpose. In this work, the observed genotypes are all ordered thus.

4.1.1 Three Elements in Linkage Analysis

There are three fundamental elements in linkage analysis: prior probability, penetrance function and segregation probability.

Prior probabilities pertain only to founders. If G is a possible genotype for a founder, then in the absence of other knowledge, $Prior(G)$ is the probability that the founder carries genotype G. Almost all models postulate that prior probabilities conform to Hardy-Weinberg (based on allele frequencies) and linkage equilibrium (may be multilocus frequencies). Here we take the uniform prior, which is 0.0162 $(1/16)$ under the 4-allele's circumstance.

Penetrance function specifies the likelihood of an observed phenotype X given an unobserved genotype G. We denote penetrance by $Pen(X|G)$. Penetrance apply to all people in a pedigree, founders and non founders alike. In general, $Pen(X|G)$ can represent a conditional likelihood as well as a conditional probability. This could be the case that $Pen(X|G)$ is some real numbers that lie in interval $[0, 1]$. The assumption we used here is full-penetrance, *i.e* $Pen(X|G)$ is either 1 or 0.

The last term is the segregation probabilities. Let $Tran(G_o|G_m, G_f)$ denote the probability that a mother with genotype G_m and a father with genotype G_f produce a child with genotype G_o. For ordered genotypes, the child's genotype G_o can be visualized as an ordered pair of gametes (U_m, V_f), U_m being maternal in origin and V_f being paternal in origin. Because any two parents create gametes independently, the segregation probability:

$$Tran(G_o|G_m, G_f) = P(U_m|G_m) * P(V_f|G_f)$$

factors into two gamete segregation probabilities.

4.1.2 Recombination

Specification of gamete segregation probabilities is straightforward for single-locus models. For a single autosomal locus, $P(U_m|G_m)$ is either 1, 0.5 or 0 and so is for $P(V_f|G_f)$, depending on whether the single allele U_m is identical in state to both, one, or neither of the two alleles respectively.

We used the Haldane's model, which postulates that recombination occurs independently on disjoint intervals. To apply Haldane's model, one begins by discarding all homozygous loci in the parent. This entails no loss of information because recombination events can never be inferred between such loci. Between each remaining adjacent pair of heterozygous loci, gametes can be scored as recombinant or no recombinant. Once adjacent intervals have been consolidated to the point where all interval endpoints are marked by heterozygous loci, calculation of gamete transmission probabilities becomes straightforward. Invoking independence, the probability of a gamete is now 0.5 times the product over all consolidated intervals of the corresponding recombination rate r or their complements $1 - r$, depending on whether the gamete shows recombination on a given interval or not. The factor of 0.5 accounts for the parental chromosome chosen for the first locus. In the exceptional case where there are no heterozygous loci, the gamete transmission probability is 1. If there is only one heterozygous locus, the gamete transmission probability is 0.5.

4.2 Likelihood Calculation

The likelihood L of a pedigree with n people can now be assembled from these component parts. Let the ith person have phenotype X_i and possible genotype G_i. Conditioning on the genotypes of each of the n people yields the representation of the likelihood:

$$\begin{aligned}\mathrm{L} &= \sum_{G_1}\dots\sum_{G_n}\mathrm{Pr}(X_1...X_n|G_1,...G_n)\mathrm{Pr}(G_1,...G_n)\\ &= \sum_{G_1}\dots\sum_{G_n}\prod_i \mathrm{Pen}(X_i|G_i)\mathrm{Pr}(G_1,...G_n)\\ &= \sum_{G_1}\dots\sum_{G_n}\prod_i \mathrm{Pen}(X_i|G_i)\prod_j \mathrm{Prior}(G_j)\prod_{(k,l,m)}\mathrm{Tran}(G_m|G_k,G_l) \qquad (4.1)\end{aligned}$$

The product on j is taken over all founders and the product on k, l, m is taken over all parent-offspring triples. Let's take a simple example as shown in Figure 3.1.

To calculate the likelihood, first we calculate the $Prior(G)$ only over 1 and 2. Then get $P(G_o|G_m, G_f)$ over the 2 nuclear family: 3, 4, 5 and 1, 2, 3. We take the full penetrance which means we could eliminate any genotypes G with $Pen(X|G) = 0$. Later on we will see that indeed we are dealing with an ideal case, where genotypes for members are all known or at least most are known so that we don't have to take phenotypes into consideration and eliminate the penetrance in likelihood immediately.

4.3 Exhaustive Algorithm

The exhaustive algorithm is considered to be a quite straightforward choice in calculating the likelihood of the data based on the pedigree. It is low efficient compared with the existing linkage algorithms and performs badly when the scale of loci and number of individuals become larger. The advantage of the exhaustive algorithm is that it is easy to implement. We use exhaustive algorithm to validate the result given by the other algorithms.

Given parental and child's genotypes, we calculate the likelihood of getting such a child from the parent. The process could be interpreted as follows:

- Input the parents' and child's genotypes; in order to simplify the input and make the judgments easy, binary data are used to denote the haplotype in each locus, where 1 stands for the dominant gene and 0 stands for the recessive gene. For example if the genotype is $AaBb$, then we transform them to 1010. Since we don't know which two alleles are on the same chromosome so we have to consider all possible combinations of the two pairs of alleles: it could be A going with b while a accompanied by B or totally different! It becomes quite tricky when more loci are under consideration

- Input the recombination rate r. Usually r lies in the interval $[0, 0.5]$ and based on the previous knowledge we mainly focus on an interval range from 0 to 0.5;

- Logical judgments are taken in the first step, *i.e* to see if it is possible for the couple under given genotypes to get such a child. This is to somehow simplify the program. In the likelihood calculation we notice that there are some genotypes that can't induce an offspring as a given child. Based on this, we eliminate the impossible parental genotypes at the beginning. For example if parents' are $AaBB$ and $AABB$, it is impossible for them to get a child whose genotype is $aaBB$ unless there is a mutation, which is not considered here.

- Output the likelihood.

Genotype	Ab	AB	ab	aB
Probability	$0.5r$	$0.5(1-r)$	$0.5(1-r)$	$0.5r$

Table 4.1: Possible egg's type and the corresponding probabilities

Genotype	AB	Ab	aB	ab
Probability	$0.25(1-r)$	$0.25r$	$0.25r$	$0.25(1-r)$

Table 4.2: Updated table for Table 4.1 under non-informative sequence.

4.3.1 Programming Interpretation

What we are going to emphasize here is the recombination rate, r. If two alleles are not in the same chromosome, the probability for the child to get the two alleles such as Ab is $0.5r$. If our data is concatenation of several chromosomes, the 'recombination' probability between neighboring loci that belong to different chromosomes becomes the probability of starting the copying of one chromosome from the opposite lineage (*i.e* paternal instead of maternal or maternal instead of paternal) of the lineage we finished copying the other chromosome from. These are believed to be independent, and the 'recombination' probability should be 0.5. Bottom line is that everything becomes simpler if you just focus on neighboring loci and the probability of recombination between them and forget about the global picture.

Assume that the mother's genotype is $AaBb$ where AB are on the same chromosome and ab are on the other chromosome, the possible egg's type and the corresponding probability is as shown in Table 4.1.

The above situation is based on the explicit information of the chromosomal sequence. Now we expand it to a more general way. Suppose that the sequence is non-informative which means we don't know whether AB are on the same chromosome or aB are on the same chromosome. In this case we take our chance by multiplying 0.5 to each probability and list all possibilities. Assume that AB are on the same chromosome and the corresponding table is shown in Table 4.2.

Suppose the father's genotype is $AaBb$ too. First we investigate in the situation where aB are on the same chromosome and we get a table for the father and combining the two parental

Mother/Father	Ab	AB	ab	aB
Ab	$0.25^2r(1-r)$	$0.25^2(1-r)^2$	$0.25^2(1-r)^2$	$0.25^2r(1-r)$
AB	0.25^2r^2	$0.25^2r(1-r)$	$0.25^2r(1-r)$	0.25^2r^2
ab	0.25^2r^2	$0.25^2r(1-r)$	$0.25^2r(1-r)$	0.25^2r^2
aB	$0.25^2r(1-r)$	$0.25^2(1-r)^2$	$0.25^2(1-r)^2$	$0.25^2r(1-r)$

Table 4.3: Possible genotypes for the child given the parents' genotypes with consideration of the ”non-informative” sequence

tables together we get Table 4.3. Table 4.3 illustrates all possible genotypes for their children and the corresponding probability.

According to each chromosome sequence, we give a transmission matrix with the multiplicator 0.25. For the father the two matrices are:

$$PF1\text{=}\begin{pmatrix} \text{0.25r} & \text{0.25(1-r)} \\ \text{0.25(1-r)} & \text{0.25r} \end{pmatrix},$$

$$PF2\text{=}\begin{pmatrix} \text{0.25(1-r)} & \text{0.25r} \\ \text{0.25r} & \text{0.25(1-r)} \end{pmatrix}$$

The mother's situation may be deduced by analogy and we could get $PM1$, $PM2$. Then we multiply the paternal transformation matrix by the maternal transformation matrix and get a child genotype likelihood matrix, *i.e* we could multiply $PF1$ by $MF1$ and the child matrix is:

$$CF1\text{=}PF1\times\text{MF1}= \begin{pmatrix} 0.25^2(r^2+(1-r)^2) & 0.25^2r(1-r) \\ 0.25^2r(1-r) & 0.25^2(r^2+(1-r)^2) \end{pmatrix}$$

Repeat this procedure and we get 4 child matrices corresponding to 16 genotypes and we could sum the probability representing the same phenotypes. Combined with the judgment step, we could get the child's likelihood. This example is based on the non-informative situation in which we don't know the chromosomal sequence while later on all of our examples are based on the informative chromosomal sequence.

4.3.2 Evaluation on Exhaustive Algorithm

We find that the exhaustive algorithm is powerless in dealing with situations which have a large number of loci; if we have n loci, we have to consider all 2^n possible haplotype and calculate $n2^{2n}$ times in order to get the likelihood in a triple. As refer to the number of member in the pedigree, it scales polynomially, which is much worse than Elston-Stewart algorithm.

4.4 Elston-Stewart Algorithm

Now we implement the Elston-Stewart algorithm to calculate the pedigree likelihood. Among all the existing algorithms for linkage analysis, the Elston-Stewart algorithm is an efficient way to deal with very large pedigrees and limited to a few markers. This algorithm performs under the Mendelian Models and is composed of three elements: prior probabilities for the founders, segregation probabilities for offspring genotypes given parents, and the last one is penetrance for individual phenotypes given genotypes. The Elston-Stewart algorithm scales exponentially in the number of loci, and linearly in the number of pedigree members. This algorithm searches the pedigree from the bottom to the top and provides a way of calculating the likelihood in a recursive manner, allowing for the possibility of computer-based linkage analysis in general pedigrees. One of the main features of this algorithm is its dependence on peeling. In the process of peeling, small nuclear families within a larger pedigree are analyzed, and all of the information is collapsed on to one of the parents(or other relatives), whose own sibship is analyzed next, and so on until all of the information is collapsed onto one final person. This method is quite straightforward unless there is a loop in the pedigree, however this situation is out of consideration here. The calculating formulation based on Elston-Stewart Algorithm is as follows:

$$
\begin{aligned}
\text{L} &= \sum_{G_1} \dots \sum_{G_n} \text{Pr}(X_1...X_n|G_1,...G_n)\text{Pr}(G_1,...G_n) \\
&= \sum_{G_1} \dots \sum_{G_n} \prod_i \text{Pen}(X_i|G_i)\text{Pr}(G_1,...G_n) \\
&= \sum_{G_1} \dots \sum_{G_n} \prod_i \text{Pen}(X_i|G_i) \prod_j \text{Prior}(G_j) \prod_{k,l,m} \text{Tran}(G_m|G_k,G_l) \qquad (4.2)
\end{aligned}
$$

4.4.1 Algorithm Description

Here is the algorithm [5]:

1. For each pedigree member, list only those ordered genotypes compatible with his or her phenotype;
2. For each nuclear family:
   (a) Consider each mother-father genotype pair.
      - Determine which zygotes can arise from the genotype pair.
      - If each child in the nuclear family has one or more of these zygote genotypes among his or her current list of genotypes, save the parental genotypes and any child genotype matching one of the created zygote genotypes.
      - If any child has none of these zygote genotypes among his or her current list of genotypes-in other words, is incompatible with the current parental pair of genotypes-take no action to save any genotypes.

   (b) For each person in the nuclear family, exclude any genotypes not saved during step $2.a$ above.
3. Repeat part 2 until no more genotypes can be excluded.

4.4.2 Examples for Elston-Stewart Algorithm

Figure 4.1 shows a simple nuclear family, constituted of two parents and a child. As can be seen, the phenotypes for mother and child are AB and Ab respectively. If we infer the father's

Person	Genotypes	Genotypes
Father	$AABB$, $AABb$, $AAbb$, $AaBB$, $AaBb$, $Aabb$, $aaBB$, $aaBb$, $aabb$	9
Mother	$AaBb$, $AABB$, $AaBB$, $AABb$	4
Child	$Aabb$, $AAbb$	2

Table 4.4: All 72 possibilities for 2 loci to consider

Person	Genotypes	Genotypes
Father	$AABb$, $AaBb$, $AAbb$, $Aabb$, $aabb$, $aaBb$	6
Mother	$AABb$, $AaBb$	2
Child	$AAbb$, $Aabb$	2

Table 4.5: 24 possibilities based on the algorithm.

genotype from top to bottom, we have to consider all 16 possible genotypes. However, if we consider the situation from bottom to up, we can infer immediately the possible genotypes for father could not be $AABB$, $AaBB$ or $aaBB$. We can give this procedure in Table 4.4 and Table 4.5. This is the basic nuclear family and 2 loci would be the simplest situation taking recombination into consideration.

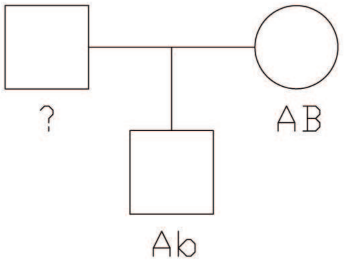

Figure 4.1: A simple example in illustrating likelihood calculation

Let's see a complicated one, an example for blood type for human beings in Figure 4.2.

In Table 4.6 and Table 4.7 we give potential genotype sets and their sizes based on iteration over all genotypes for the 9 individuals according to the 3 ABO alleles is $6^9 = 10077696$, which is a fairly large set! While conditional on phenotype, we find that the calculation could

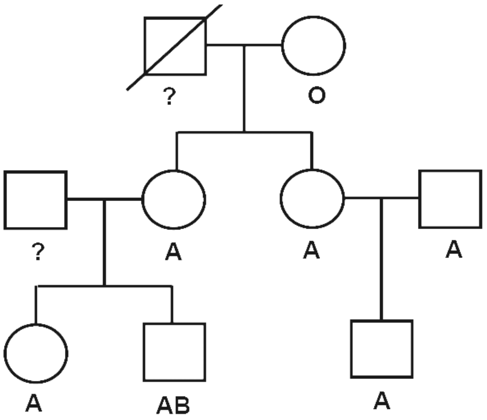

Figure 4.2: A pedigree Partially Typed at the ABO locus

Person	Genotype	Genotype
$I-1$	AA, AO, BB, BO, AB, OO	6
$I-2$	OO	1
$II-1$	AA, AO, BB, BO, AB, OO	6
$II-2$	AA, AO	2
$II-3$	AA, AO	2
$II-4$	AA, AO	2
$III-1$	AA, AO	2
$III-2$	AB	1
$III-3$	AA, AO	2

Table 4.6: 1152 possibilities to consider

be accelerated remarkably. This example shows a big advantage of using the Elston-Stewart algorithm in a pedigree.

There is a classic software called LINKAGE which implements the Elston-Stewart algorithm in it. We implement the Elston-Stewart algorithm in C programming language and give examples to illustrate it.

Take a single nuclear family as an example. Suppose that we have a segment of gene from a child and the parents' genetic segments. We want to calculate the probability of getting such a child's gene based on the parental genetic segments. Because our data is concatenation

Person	Genotype	Genotype
$I-1$	AA, AO, AB	3
$I-2$	OO	1
$II-1$	BO, AB	2
$II-2$	AO	1
$II-3$	AO	1
$II-4$	AA, AO	2
$III-1$	AA, AO	2
$III-2$	AB	1
$III-3$	AA, AO	2

Table 4.7: 48 possibilities based on the algorithm

of several chromosomes, the ”recombination” probability between neighboring loci that belong to different chromosomes becomes the probability of starting the copying of one chromosome from the opposite lineage (*i.e.* paternal instead of maternal or maternal instead of paternal) of the lineage we finished copying the other chromosome from. These are believed to be independent, so here the ‘recombination’ probability should be 0.5. Then we consider the child’s following locus: if this gene is not from the same chromosome as the previous one, *i.e* there is no cross over, then we multiply 0.5 by a recombination rate, r, between these two loci; otherwise we multiply 0.5 by $(1-r)$. Notice that there are some situations in which the existing child’s genetic segment can choose from both parental chromosomes, which is to say the parent is homozygote at this locus, we need to sum both recombinant and non-recombinant situation based on the last locus; we will illustrate this procedure in the flow chart below. In order to mark the source of each allele in the child’s segments, we use an indicated variable. An indicated variable is a binary variable takes 1 or 0, where 1 denotes there is a recombination and 0 indicates there is no recombination.

Another problem arises: how do we know the origins of each chromosome from the child. Actually we don’t know so we have to think thoroughly to make sure all situations are considered. Let’s call the probability we get above triplet(G_c,G_m), so the probability based on this

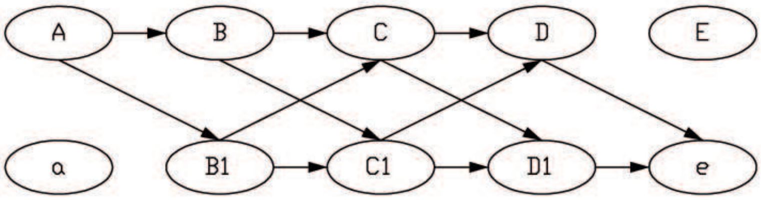

Figure 4.3: How the Elston-Stewart algorithm goes through the loci; achieved segment been marked as $ABCDe$

nuclear family becomes:

$$Prob = 0.5 * (triplet(G_{c_1}, G_m) * triplet(G_{c_2}, G_f) + triplet(G_{c_2}, G_m) * triplet(G_{c_1}, G_f))$$

Prior probability is another important thing. This is to define the probability in getting such a pair of parents. Suppose equally frequency, 0.5, for each allelic type so that we have half chance to observe recessive or dominant in a locus from the population. In this situation non-informative priors are better choices and we consider uniform prior in a discrete sample space. Suppose there are N loci, then the sample space has exactly 2^{2N} available sample points, so we take the prior probability to be 1/2^{2N}.

We could describe the crossover process in a mimic way, using a sketch map in Figure 4.4.

From this flow chart we easily see that the Elston-Stewart algorithm performs badly when there are many loci, say, 20; because it scales exponentially in the number of loci, the time complexity should be O(2^n).

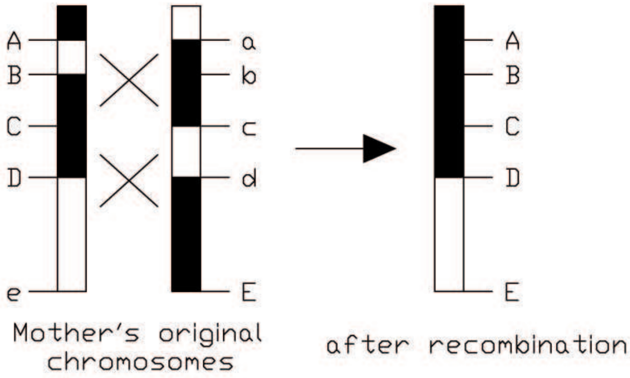

Figure 4.4: Generation of a child's chromosome. The crosses within mother's chromosomes mean there are recombinant events happened.

4.5 Lander-Green Algorithm

There are other algorithms which are widely used as well and the famous one is the Lander-Green algorithm proposed in 1987. Even with complete phenotyping of all pedigree members, phase ambiguities pose a problem. Lander and Green took a different approach. They redefine the likelihood expression so that the sums extend over loci rather than people. In other words, their algorithm steps through the likelihood calculation locus by locus while considering all people simultaneously at each locus.

Although there are some speedups, very large pedigrees are simply beyond the reach of the Lander-Green algorithm. Above all, there is no such an algorithm that could search the pedigree linearly both in the number of members and the number of loci. The Lander-Green algorithm is another approach for getting the pedigree likelihood. It scales linearly in the number of loci. Since all pedigree members are taken simultaneously, it scales exponentially in the number of individuals.

Since the method of Lander and Green shifts summations from people to loci so it is possible to decompose on both people and loci in such a manner that the prior, penetrance, and transmission arrays factor. This suggestion entails viewing the multilocus ordered genotypes of a given person as originating from a Cartesian product of his or her single-locus ordered genotypes. A negative consequence of this synthesis is the substitution of a swarm of small arrays where a few large ones formerly sufficed. In compensation for this complication is the potential benefit of encountering much smaller initial and intermediate arrays in the likelihood calculation.

That is to say that the probability of observing the allele v at locus $k+1$, given the alleles at locus k, is independent of either the value of the alleles or the genotype at any locus to the left of locus k. This Markov property could be used in the Lander-Green algorithm and yields a transformed Markov Model, the Hidden Markov Model (HMM). HMM is a statistical model in which the system being medelled is assumed to be a Markov process with unknown parameters, and the challenge is to determine the hidden parameters from the observable parameters. The extracted model parameters can then be used to perform further analysis.

In a regular Markov model, the state is directly visible to the observer, and therefore the state transition probabilities are the only parameters. In an HMM, the state is not directly visible, but variables influenced by the state are visible. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. A quite straightforward way in explaining the HMM is shown in Figure 4.5.

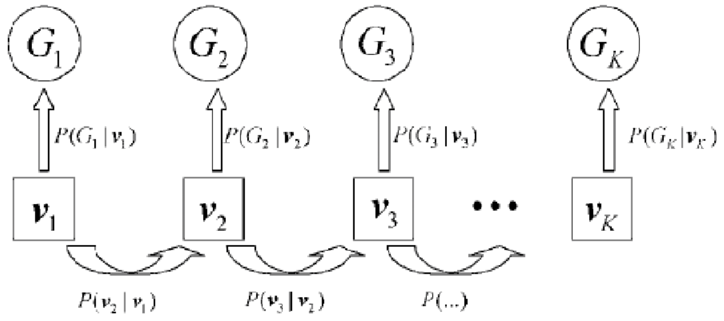

Figure 4.5: HMM used in the likelihood calculation.

In order to differ Lander-Green algorithm from Elston-Stewart algorithm, we will use another group of notations to calculate the likelihood. As shown above, there are three components in the likelihood calculation based on Lander-Green algorithm: one ingredient is the observed genotype X_i for each locus; a second ingredient will be the possible allele at each locus, $P(X_j|I_j)$; the final ingredient connect the allele along the chromosome, $P(I_{j+1}|I_j)$. I variables represent the "inheritance variable". We only consider multi point phase under full penetrance and later all genotypes given in our examples are given based on chromosome.

Like a general Markov Chain, finding the transition matrix is of key point. In order to get the transition probability, recall the transition matrix in Markov Chain. A two-state Markov Chain (X_t) $t = 0, 1$...has transition matrix: P=$\begin{pmatrix} 1-\alpha & \beta \\ \beta & 1-\alpha \end{pmatrix}$

From the property of Markov Chain, we have: $P^{n+1} = P^n P$. Here we change the entries in the matrix into recombination fraction, so with one meiosis, our transition matrix is:

T=$\begin{pmatrix} 1-\theta & \theta \\ \theta & 1-\theta \end{pmatrix}$

Meiosis is important in the Lander-Green algorithm since it's the basic calculating unit in the formula. With two meiosis,the transition matrix is:

$$T^{\otimes 2}=\begin{pmatrix} (1-\theta)T & \theta T \\ \theta T & (1-\theta)T \end{pmatrix} = \begin{pmatrix} (1-\theta)^2 & (1-\theta)\theta & (1-\theta)\theta & \theta^2 \\ (1-\theta)\theta & (1-\theta)^2 & \theta^2 & (1-\theta)\theta \\ (1-\theta)\theta & \theta^2 & (1-\theta)^2 & (1-\theta)\theta \\ \theta^2 & (1-\theta)\theta & (1-\theta)\theta & (1-\theta)^2 \end{pmatrix}$$

The recursive formulation for n meiosis is:

$$T^{\otimes n}=\begin{pmatrix} (1-\theta)T^{\otimes(n-1)} & \theta T^{\otimes(n-1)} \\ \theta T^{\otimes(n-1)} & (1-\theta)T^{\otimes(n-1)} \end{pmatrix}$$

In general, the transition matrix is patterned and the transition probability depends on two things: one is the number of meiosis that were outcome changed; the other is the number of meiosis that were outcome did not change. The entries in the transition matrix are the products of powers of θ and $1-\theta$. The size of the transition matrix is $2^n \times 2^n$ for n meiosis, *i.e* n triples which are quite large. The formula for likelihood calculation in Lander-Green algorithms is as follows:

$$\mathrm{L} = \sum_{I_1} \dots \sum_{I_m} \mathrm{P}(I_1) \prod_{i=2}^{m} \mathrm{P}(I_i|I_i - 1) \prod_{i=1}^{m} \mathrm{P}(G_i|I_i) \tag{4.3}$$

where $P(I_1)$ is the prior probability of allelic states along the chromosome; $P(G_i|I_i)$ is the probability of observed genotypes, given the allelic states; and $P(I_{i-1}|I_i)$ is the model for transitions along chromosome to the allelic states, which depends on the recombination fraction. It combines with Bayesian theory so that it can estimate probability of each allelic state at each locus. Generally it is quite slow unless there are few loci, but Lander-Green algorithm can scale the pedigree in a linear time according to the number of loci.

4.5.1 Flow of Lander-Green Algorithm

This is the Lander-Green Algorithm:

1. Enumerate all possible ”inheritance vectors” in the input pedigree given n non-founders. This is done by listing all meiosis in the pedigree: there should be $2n$ meiosis for n non-founders (each non-founder has two parents so n non-founders implies $2n$ meiosis);

2. Iterate over inheritance vectors and markers to calculate the probability of the observed genotypes for each marker conditioned on a particular inheritance vector: $P(X_i|I_i)$. This is done using the ”genetic descendant graph”, as introduced in the previous chapter. List all possible IBD patterns: total of $22n$ possible patterns defined by setting each meiosis to one of two possible outcomes;

3. Build transition matrix for moving along chromosome: patterned matrix, built from matrices for individual meioses;
$$T^{\otimes n} = \begin{pmatrix} (1-\theta)T^{\otimes(n-1)} & \theta T^{\otimes(n-1)} \\ \theta T^{\otimes(n-1)} & (1-\theta)T^{\otimes(n-1)} \end{pmatrix}$$

4. Run the Markov chain: start at first marker, $m = 1$ and then build a vector listing $P(G_{firstmarker}|I)$ for each I; move along chromosome and then multiply vector by transition matrix combined with information at the next marker; also multiply each component of the vector by $P(G_{currentmarker}|I)$. Repeat previous two steps until done.

An obvious advantage of Lander-Green algorithm is that it could be implemented from wherever from the chromosome based on the Markov property: such as forward recurrence, backward recurrence or even arbitrary location. Since all the genotypes are known, we could eliminate the terms where $P(G_i|I_i)$ are all 1. As mentioned before, the Lander-Green algorithm

	Sib1	Sib2
Locus A	1/1	2/2
Locus B	1/1	1/1

Table 4.8: A basic example in Lander-Green algorithm

I_1	I_2	$P(I_1)$	$P(I_2 \vert I_1)$	$P(X_1 \vert I_1)$	$P(X_2 \vert I_2)$	L
0	0	0.25	0.67	0.0625	0.0625	0.00067
0	1	0.25	0.30	0.0625	0.125	0.00058
0	2	0.25	0.03	0.0625	0.25	0.00013
1	0	0.5	0.15	0	0.0625	0.00000
1	1	0.5	0.70	0	0.125	0.00000
1	2	0.5	0.15	0	0.25	0.00000
2	0	0.25	0.03	0	0.0625	0.00000
2	1	0.25	0.30	0	0.125	0.00000
2	2	0.25	0.67	0	0.25	0.00000

Table 4.9: Output of Lander-Green algorithm for two siblings under Hardy equilibrium

searches the pedigree locus by locus so for each marker in an offspring, the algorithm searches throughout his ancestors and produce a possible inheritance vector.

Now let's see some examples based on Lander-Green algorithm. First of all we introduce a definition called IBD which stands for identical by descend. It means that two alleles have a common ancestor before the start of the population. This is the underlying sharing of chromosomes segregating within a family, e.g the siblings sharing 0, 1 or 2 alleles, with probability 0.25, 0.5 and 0.25 respectively. The following example considers two loci separated by recombination fraction, $\theta = 0.1$; and each locus has two alleles, both with frequency 0.5, the prior probability.

Suppose the two siblings have the genotypes shown in Table 4.8.

Consider the Lander-Green algorithm, we obtain the Table 4.9 of the probability of $IBD = 2$ at marker B when we consider B alone and consider both markers simultaneously.

4.6 Comparison between Elston-Stewart Algorithm and Lander-Green Algorithm

Now we could give a comprehensive comparison between the Lander-Green algorithm and the Elston-Stewart algorithm. They are both for multi-point analysis and very good choices in researching pedigree data. The running time for Lander-Green algorithm is of the order $O(m2^{4m})$, where m is the number of loci under investigation and n is the number of nonfounders. Later Elston and Indury improved the algorithm and their version runs in time $mn2^{n-2}$.

The major difference between the two algorithms is the complexity.The complexity of the Elston-Stewart algorithm scales exponentially in the number of loci and linearly in the number of people, and the Lander-Green algorithm scales exponentially in the number of people and linearly in the number of loci. There are also differences in the types of analysis each can handle. In general, the Elston-Stewart algorithm is more flexible in the models since it uses multilocus genotypes. On the other hand, the Lander-Green algorithm uses an HMM that computes the likelihood adding a single locus at a time making it more difficult to handle haplotype frequencies and adding multilocus models increases the computational complexity significantly. In respect of the missing data, the Lander-Green algorithm is preferable.

4.7 The Incremental Algorithm

The incremental algorithm is an algorithm for computing the convex hull of a set of points in two or more dimensions. The basic idea is to add points one at a time updating the hull as we proceed. In the linkage analysis, the incremental algorithm could be used to get the likelihood from a contrasting pedigree. We can construct a contrasting pedigree to a pedigree under research by adding some individuals into the pedigree or changing some information such as some genotypes. We focus on a special situation in which all genotypes are known for each individuals in the pedigree.

Each time we insert some new member or change some genotypes, we update the likelihood we already got by changing it 'locally'. First of all we save the probabilities triple by triple, which is to say we store each of them in a ’family space’ instead of as a apart of the whole pedigree; then label the individuals which are directly dependent to the new insert and identify them as triples, if there is a ’half-family’ inside our labeled individuals we update this ’half-family’ by adding another triplet ancestor to it: for example if the mother and son are taken then we just add the father; finally we update those labeled individuals triple by triple and multiply them by the ’untouched’ ones.

The genotype networks are used in the incremental algorithm. Let’s take Figure 3.3 as an example. If we change the genotype for individual 5, the update issues are done to families 3, 5, 10 and 5, 8, 12. On the other hand if we insert a sibling for 3, we update the likelihood by multiplying it by a probability for the ancestor 1 and 2 to get such a sibling.

4.8 Simulations

4.8.1 Simulation Based on the Elston Stewart Algorithm

In this part we implement the Elston-Stewart algorithm and the Lander Green algorithm and use them to calculate the probability of the data giving the pedigree, which is used as the ’likelihood’ of the pedigree. Simulations for the incremental algorithm is given in the form of comparison two pedigrees’ likelihood. The incremental algorithm is a better way in choosing a more ’probable’ pedigree since it only accounts for the variety. All simulations used in this work are based on specified sequence, which is to say the genotype is fully observed not only for each allelic type but also the alleles’ positions are known.

We try some typical cases under 2 loci and get their probabilities respectively based on the Elston-Stewart algorithm. Since 2 loci, simple nuclear family is the fundamental thing we have to be familiar and confident with. On the other hand, exhaustive algorithm is an option in proving the result from the Elston-Stewart algorithm. The exhaustive algorithm is used here to prove the results based on the Elston-Stewart algorithm and we are happy to find the results match with each other.

We find that the exhaustive algorithm is powerless in dealing with situations which have a large number of loci; if we have n loci, we have to consider all 2^{2n} possible genotypes and calculate $16n^4 2^{2n}$ times in order to get the likelihood in a triple. To the number of member in the pedigree, it scales polynomially, much worse than Elston-Stewart algorithm.

4.8.2 Examples

- Given the child's and parents' genotypes, shown in Figure 4.6, calculate the likelihood for this pedigree. We take discrete uniform distribution with each allele with frequency 0.5 as the prior probability. This is the general prior probability we choose unless otherwise declared. Alleles in the same row are in the same chromosome. Unless otherwise specified, all recombination rates are taken as 0.25.

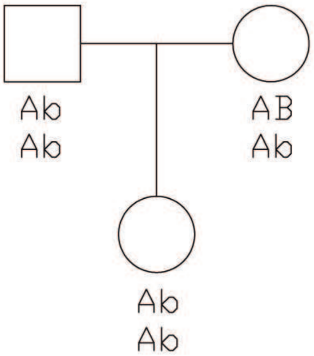

Figure 4.6: Single nuclear family, 2 loci

According to the Equation 4.2 The corresponding likelihood based on the pedigree is 9.7656×10^{-4}.

- Given parents genotypes, we want all children's genotypes and their corresponding probabilities, as shown in Figure 4.10:

We get an output table, shown in Table 4.10

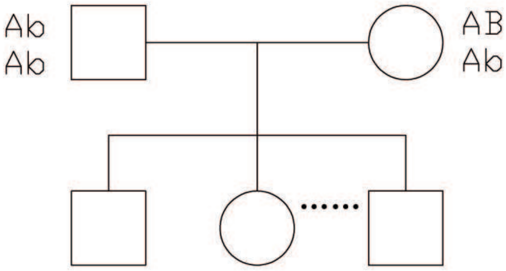

Figure 4.7: Generating all possible children.

Genotypes	Likelihood
AABb	1.4648×10^{-3}
AaBb	9.7656×10^{-4}
AAbb	9.7656×10^{-4}
Aabb	1.4648×10^{-3}

Table 4.10: Possible offsprings for the parents. Note that the maternal haplotype must be Ab since mother is homozygous in both loci.

If we consider this case under original Mendelian Model, which is under the assumption of independent heredity, we will get the four genotypes equally with probability 0.25.

- Figure for this example is like the one we illustrated above in Figure 4.6, the nuclear family style. Now we change the situation: only the mother and child's genotypes are known. There is a 'switch' which can give us two approaches in leading to it: one is by calculating the possibility for each haplotype obtained from mother then sum them and dividing the result by 2; the other way is complicated: we have to generate all possible genotypes for the father and then calculate them in turn. Although it is not easy, the advantage is that we can have every genotype with its corresponding likelihood. We will explain how to inference father's genotypes' space later. The result is 1.9531×10^{-3}.

- Given a pedigree consisting of three generations (like Figure 4.8): grandparents in the father's side, parents, and one child. The recombination rate is 0.186 and the father's genotype is missing.

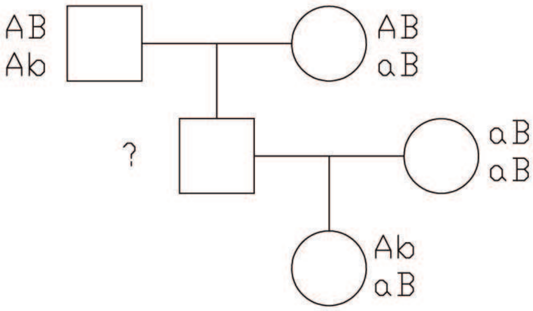

Figure 4.8: 3-generation pedigree, 2 loci.

  We want to get the likelihood for this pedigree. Thinking from the grandparents, we get 4 genotypes with equal probabilities; while considering from the bottom, we find that only one genotype is possible in this pedigree. The probability for getting such a child from the data based on the pedigree is 5.6763×10^{-6}.

- Take Figure 4.8 as another example in which we remove grandfather's genotype. In this situation two people are 'missing' and we peel the pedigree from the bottom to the top and each time we inference some potential genotypes for the missing data, *i.e* peel the bottom family first. The calculated result is 1.2813×10^{-3}.

- For a two generation pedigree with more than 1 child and unknown father's genotype, for example in Figure 4.9, the probability is 5.1997×10^{-4}.

- The last thing we concerned based on 2 loci is three generation with several offsprings in the bottom. Parental genotypes are given in Figure 4.10.

  The likelihood for this pedigree is 0.

More complicated examples are given: more members, more loci.

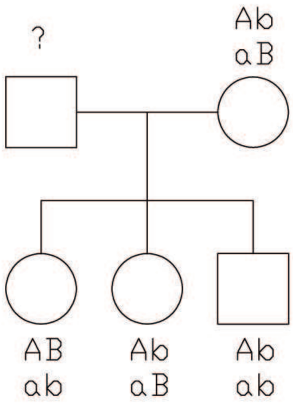

Figure 4.9: Single nuclear family, 3 children, 2 loci

1. Consider a pedigree shown in Figure 4.11. This is an example based on 6 loci, 5 individuals with missing data at locus 5 from the paternal side and at locus 6 from the maternal gene. In this situation we sum up all the possible genotypes for the missing position and consider that some allelic combinations are not suitable in this situation: it is impossible for the father to hold 2 dominant alleles at locus 5. If we consider discrete uniform distribution with equal frequency for each allelic type, we assign each entry with a weight $1/2^{12}$. The likelihood based on this situation is 6.1483×10^{-16}.

2. Another example is a larger pedigree shown in Figure 4.2 with genotypes observed for all the individuals in the pedigree. Their observed genotypes are given in Table 4.11, the recombination vector is $r = (0.23, 0.15, 0.18, 0.35, 0.27, 0, 32)$. The final likelihood we get is 8.4382×10^{-22}.

3. We can not only deal with the situation with partial missing value but also situations in which a person's genotype is totally unknown. Take Figure 4.12 as an example and in this situations the likelihood is 1.2207×10^{-4}.

4. In this example, we use incremental algorithm to compare two pedigree. We emphasize again the 'likelihood' used throughout this work is an expression as 'the probability of the data given the pedigree'. Refer to the aspect of 'likelihood', we need some benchmark, or another model to compare and according to certain indices we can tell which pedigree is more likely to happen. Suppose a special situation within which the haplotype's prior

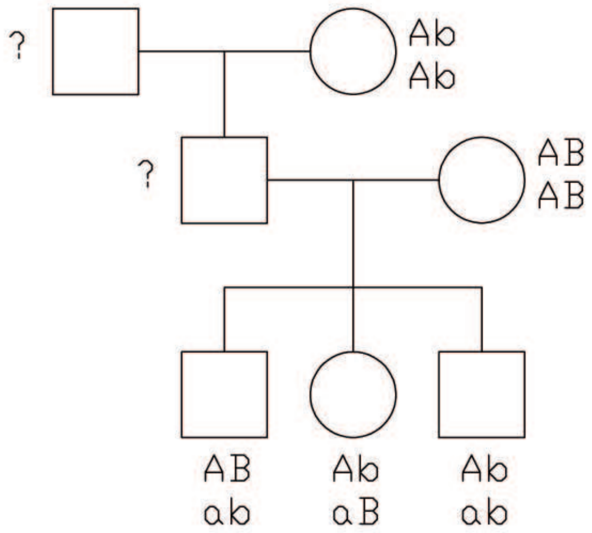

Figure 4.10: 3-generation pedigree with some missing data.

probability for the founder is $P(0, 0...0) = P(1, 1...1) = 0.5$, which is to say for each founder, all the alleles in a chromosome are either recessive or dominant and no mixture. In the pedigree comparison, we want to get the likelihood for getting a particular child based on two pedigree: one is a nuclear family with the parents as the founders while the other is a three generation pedigree in which the grandparents and the father are the founders. Like the hypothesis test in the statistics, we express this problem in Figure 4.13. The only observed genotype is from the child, whose gene sequence is $1010/0110$ on each chromosome. Suppose the recombination rate vector is $r = (0.25, 0.35, 0.18)$. Results are shown in Table 4.12.

We can see that in this situation, the second pedigree will have a higher probability, which means it is more likely to produce such a child in a 3-generation pedigree under which there is enough time for each chromosome to get 'mixed'. We can't simply compare the result here with the previous examples since the assumptions for the priors are quite different.

5. In the last example the incremental algorithm is used again to get the likelihood for two pedigrees. The pedigree is like the one we use in Figure 3.3 with the corresponding

Person	Genotype
I-1	1100101/0101011
I-2	1010010/1100101
II-1	1011001/0111101
II-2	1101111/1000010
II-3	0101111/1010101
II-4	1101101/0011000
III-1	1011001/1101111
III-2	0011001/1101010
III-3	0101101/0001000

Table 4.11: Example based on the previous blood type figure. Digits lying on the same side along the forward slash denote the alleles in the same chromosome.

	Pedigree 1	Pedigree 2
Likelihood	5.6×10^{-5}	3.5156×10^{-4}

Table 4.12: Results for Figure 4.13

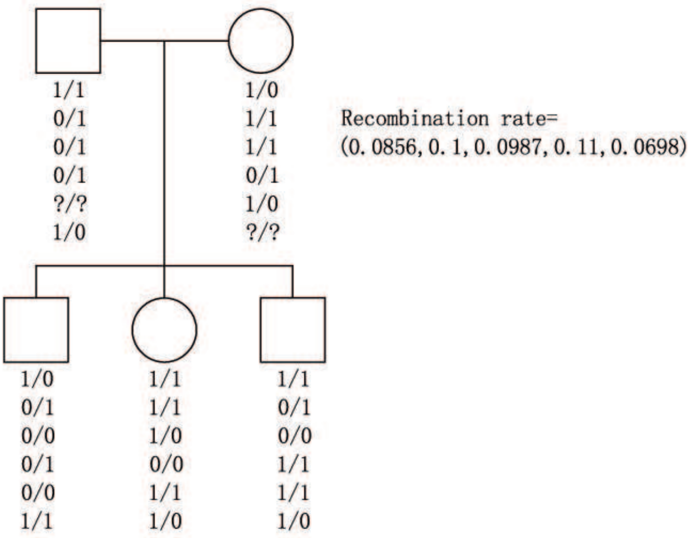

Figure 4.11: A two-generation pedigree consists of 5 members.

genotypes listed in Table 4.13.

The recombination rate r is $r = (0.125, 0.18, 0.25, 0.35)$. The likelihood for this pedigree based on these genotypes is 6.5408×10^{-30}. Imagine that we change 5's genotype into $01111/11010$ and the updated likelihood based on the incremental algorithm is 6.1041×10^{-30}. The first situation is more likely to be happened but not significantly.

For the Lander-Green algorithm we run the same example and we are happy that both programmes perform excellent and the results are the same as shown.

4.9 Generating Genotypes in a Pedigree

Suppose a nuclear family with unknown father's genotype. We name the child's two chromosome 1 and 2. Suppose 1 is from the maternal side and 2 is from the paternal side, or vice versa. First judgements are given to see if it is possible for the mother to give an egg with such haplotype. Imagine that 1 could be given by mother, label it as $1'$ and delete it. Now here comes the trick: the father has to pass a sperm with the haplotype which is the same as 2, possibly with recombination to his son. In the next step we will generate some possible father's genotype

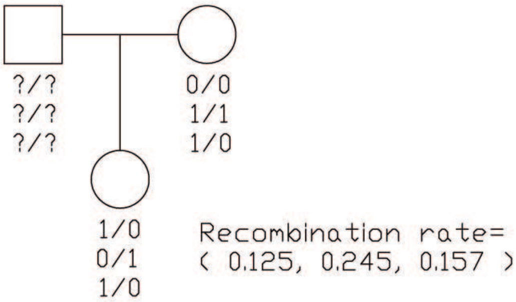

Figure 4.12: More loci with missing data.

forms, in mathematics they are just some $n \times 2$ matrices with n being the number of loci.

In order to get the possible father's genotypes, we first assign each term in 2 chromosome to each row in a $2 \times n$ matrix. Since there are 2^n potential sequences for the chromosome 2 so in this step we get 2^n matrices, while in each matrix, there are n empty seats so there are another 2^n-allele combinations for it. Thus in the end, for the father we can generate a series of 2^{2n} genotype's matrices for the father.

This is used mainly in the Elston-Stewart algorithm and indeed we use the same way in generating genotypes for the missing data in the Lander-Green algorithm.

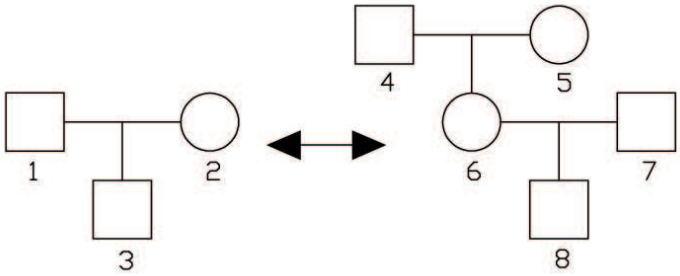

Figure 4.13: Comparison of the likelihood based on two pedigree to see which is more suitable.

Individual	Genotype
1	11010/01110
2	11001/00101
3	11010/01001
4	11111/01011
5	01000/10010
6	00001/11111
7	01011/11000
8	10011/01010
9	11110/00001
10	10010/01111
11	10000/01001
12	10001/10010

Table 4.13: Genotypes for the members in pedigree shown in Figure 3.3

Chapter 5

Conclusion

We have illustrated several methods for the linkage analysis based on calculating the likelihood of the data giving a pedigree. Notice that this is sometimes not the real situation so contrasts are suggested in identifying pedigrees. While in this situation, the incremental algorithm helps us to simplify the calculation so that we don't have to repeat doing what we have already done and needn't change. We have studied the two main algorithms in use currently: the Elston-Stewart algorithm and the Lander-Green algorithm. The exhaustive algorithm is used to validate their results. We compared them in many ways such as the time complexity and the algorithm principle. We have presented also some usages of the algorithms, and a method to generate unknown data based on Elston-Stewart algorithm. It focuses on using a haplotype from the unknown person's offspring to generate a sample space for the person. Then we search through the whole sample space by either the Elston-Stewart algorithm or Lander-Green algorithm to get the probability based on each situation. If we treat different possible situation as different pedigree, the probability we get becomes the likelihood and by comparing the likelihood we infer the pedigree with a larger probability would be a more possible one. The comparison of the likelihood is purely mathematical comparison.

Based on the simulated examples, we find the following properties for the likelihood based on a pedigree:

- The likelihood for a pedigree decreases with the number of the individuals in the last generation, *i.e* the more people we have at the bottom, the lower the probability will be;

while if founders are induced, it is hard to say how the likelihood will be;

- The more loci we have, the lower the likelihood is no matter what the recombination rates are. No doubt we are the children of fortune among the war in mother's womb!
- As can be seen from the formulation for the likelihood calculation, the prior is an important factor we should take into account. For example in the early examples we take uniform priors while in the last one a special prior is used so the likelihood is incomparable if the priors are different.

Characteristics of each algorithm become obvious when we simulate the pedigrees in the two algorithms. The Elston-Stewart algorithm keeps on peeling families off and gets the summation over persons. So the number of individuals is not a problems in this algorithm. While the Lander-Green algorithm searches through each locus over all the pedigree in turn and keeps on tracking back until get the first founders for the locus. The simulations show the Elston-Algorithm performs well under huge pedigree while poor in the number of loci, even 20 is too large for it. Quite opposite, the Lander-Green algorithm, although with several enhancements, works perfect in a pedigree with millions of loci but number of individuals is a limitation and even a 10 persons pedigree can lead it to a serious situation.

We have explained the usage of graphical models and probability networks in the linkage analysis. They can be used in formulating, analyzing and visualizing many problems in genetics, or problems having a strong genetic component. The merits of phrasing these problems in the language of graphical models derive from the flexibility with which standard problems can be modified to accommodate special situations, whether they be observational schemes, types of problem under consideration or other external circumstances that need to be incorporated into the basic genetic models [6]. So analysis of complex genetic data can benefit enormously from current rapid developments in the area of graphical models and computers.

There are several directions in which the analysis could be extended. One is through Bayesian estimation procedure to give some benchmarks with which we can compare our results since the results we get are not straightforward in telling whether this is the real pedigree or not: we can't compare it with 1 or other constance since under certain situations the likelihood tends to be infinite, for example in a pedigree with loop. Another approach is using genome-wide genetic data to reconstruct the entire pedigree that connects a set of individuals from the same population or species (Hein, 2004). In this way, we can pick the ”most probable” pedigree among all possible ones and research the pattern of inheritance based on this pedigree since in some circumstances this can be seen as the ”real” pedigree. The most attractive direction for us should be the incremental algorithm and how to accelerate them. We are aiming to find a better algorithm that can be used in fairly large pedigree and performs ideally in the increasing number of loci and individuals. Accompany the algorithms with Bayesian estimation will undoubtedly lead us to an attractive world.

Bibliography

[1] Concalo R. Abecasis and Yu Zhao. Algorithmic improvements in gene mapping. 2007.

[2] Chris Cannings and Alun Thomas. Genealogies: algebras and graphs. 2006.

[3] Anna Ingolfsdottir and Daniel Gudbjartsson. Genetic linkage analysis algorithms and their implementation. 2007.

[4] Claus Skaaning Jensen and Augustine Kong. Blocking gibbs sampling for linkage analysis in large pedigree with many loops. *Am.J.Hum.Genet*, 65:885–901, 1999.

[5] Kenneth Lange. *Mathematical and Statistical Methods for Genetic Analysis*. Springer, New York, second edition, 2002.

[6] S. L. Laruritzen and Nuala A. Sheehan. Graphical models for genetic analysis. *Statistical Science*, 18:4,489–514, 2003.

[7] S.L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.

[8] Steffen L.Lauritzen Robert G. Cowell, A Philip Dawid and David J. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Springer, London, first edition, 1999.

[9] Eric S.Lander and Philip Green. Construction of multilocus genetic linkage maps in humans. *Genetics*, 84:2363–2367, 1987.

[10] Ao Yuan and George E.Nonney. Two new recursive likelihood calculation methods for genetic analysis. *Human Heredity*, 54:82–98, 2002.