Deterministic Calculation and Stochastic Simulation in Multi-point Linkage Analysis

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Technical Sciences

presented by
MATTHIAS SCHELLING
Dipl. Ing.-Agr. ETH
born May 9, 1970
citizen of St. Gall and Berneck

accepted on the recommendation of
Prof. Dr. G. Stranzinger, examiner
Dr. C. Stricker, co-examiner
Prof. Dr. R. L. Fernando, co-examiner
Prof. Dr. N. Künzi (Supervisor), co-examiner

2004
Acknowledgments

I would like to greatly acknowledge:

Prof. em. Dr. N. Künzi for the assignment of the research topic, for his confidence and for his always kind and valuable support as supervisor and co-referee of this thesis.

Prof. Dr. R. L. Fernando for his great scientific support, for his invaluable support in understanding and programming MATVEC, for the very prosperous collaboration in Ames and in Zurich and for accepting the task as co-referee for this thesis.

Dr. C. Strieker for being my supervisor and his close support throughout the whole thesis, even after the Animal Breeding Group ceased to exist due to the retirement of Prof. Künzi, and for accepting the task as co-referee as well as for many philosophic discussions which have been an invaluable support.

Prof. Dr. G. Stranzinger for accepting the task as referee due to the retirement of Prof. Künzi and for accepting the thesis.

All my colleagues of the former Animal Breeding Group of the Institute of Animal Science at the Swiss Federal Institute of Technology (ETH) in Zurich for their assistance and for having a good time.

This project was financed by the Swiss National Science Foundation (grant 31-55953.98).
iv
Abstract

In the first part of the thesis the limitations of the Elston-Stewart algorithm to calculate the likelihood in complex pedigrees are outlined together with iterative peeling (Wang et al., 1996) to approximate likelihood calculations in such situations. Then, the Metropolis-Hastings and the Gibbs sampler as Markov Chain Monte Carlo methods (MCMC) are introduced with their properties to sample genotypes in general pedigrees. Strategies to sample certain genotypes jointly as blocks are discussed with the most extreme of such samplers: the ESIP sampler of Fernández et al. (2001). Then, sampling allelic origin known as segregation indicators by a Metropolis-Hastings sampler (Thompson, 1994; Sobel and Lange, 1996) is explained in theory and illustrated by examples.

When only a single locus is considered, the examples illustrate that for sampling genotypes the ESIP-sampler is most efficient, as it samples the genotypes of the whole pedigree jointly conditional on data. It becomes approximate only for large and complex pedigrees where the cut-set size becomes too large and thus the pedigree cannot be peeled exactly. As the ESIP-sampler samples genotypes conditional on data, it is most efficient, i.e. has a low rejection rate in general. Still considering only a single locus, the original Sobel & Lange (S&L-) sampler is less efficient, as it does not sample conditional on data with a blocking strategy that does not guarantee irreducibility unless the probability for an unlimited number of transition steps per newly proposed descent graph is non-zero. Such a non-zero probability overcoming vertical dependence, i.e. dependence between different individuals’ segregation indicators is implemented by sampling the number of transitions from a geo-
metric distribution with mean 2. Although the state space of segregation indicators may be potentially smaller compared to the state space of genotypes, the S&L-sampler did not reveal faster mixing of the Markov Chain in general due to these two drawbacks.

In the second part of the thesis multiple loci are considered. Then, the ESIP-sampler becomes increasingly (exponentially) inefficient, as multiple loci need to be peeled. Thus, for more than 2-4 loci, ESIP becomes infeasible. Applying the S&L-sampler to each locus separately in multi-locus problem may reveal extremely slow mixing for tightly linked loci. Thus two new approaches sampling segregation indicators at multiple loci non-independently were derived: the haplotype sampler and the cascading origin sampler. Both are Metropolis-Hastings samplers showing extremely good mixing properties in certain situations prevalent when loci are tightly linked. They were both capable of overcoming specific strong horizontal dependence problems, i.e. dependence between linked loci within a haplotype in certain situations. Combining them with a multi-locus version of the original S&L-sampler revealed fast and efficient mixing in all example presented here. All samplers together with deterministic maximum likelihood are implemented in MATVEC, a C++ program package especially suitable for genetic analysis in animal breeding. All multi-locus samplers presented are illustrated and tested on various examples. These new samplers efficiently calculate the probability of descent for a certain QTL-allele. Most QTL mapping approaches currently in use in animal breeding or human genetics rely on such information. Even more, it further allows to calculate the covariance matrix at a marked QTL to be used for marker assisted best linear unbiased prediction (BLUP) and thus marker assisted selection in livestock.
Zusammenfassung

Deterministische Berechnung und stochastische Simulation in Mehrpunkt-Kopplungsanalysen


Genotyp-Verteilungen, welche nicht mit den Daten verträglich sind. Dieser Sampler ist theoretisch nicht reduzierbar (irreducible), d.h. alle mit den Daten kompatiblen Realisatio-

nen von Segregationsindikatoren können vom Sampler vorgeschlagen werden. Trotzdem ist es möglich, dass Abhängigkeiten zwischen Segregationsindikatoren verwandter Individuen (vertikale Abhängigkeit) die Wahrscheinlichkeit, dass eine gewisse Realisation von Segre-
gationsindikatoren gezogen wird, beinahe Null wird. Dies, zusammen mit dem Ziehen der Stichproben unabhängig von den Daten, führt zu einer schlechten Effizienz dieses Samp-

lers.

Im zweiten Teil werden die vorliegenden Ansätze anhand von Modellen mit mehreren Loci verglichen. Die Effizienz des ESIP-Samplers nimmt mit zunehmender Zahl der Loci exponen-
tiell ab. Bereits mit 2 - 4 Loci wird dieser Sampler in größeren Pedigrees unbrauchbar. Auch der S&L-Sampler wird bei mehreren Loci ineffizient, besonders wenn die Loci eng gekoppelt sind (horizontale Abhängigkeit).

Deshalb wurden zwei neue Metropolis-Hastings Sampler für Segregationsindikatoren vor-

*MATVEC* ist in der Programmiersprache C++ geschrieben und besonders auf die Anfor-
derungen genetischer Analysen in der Tierzucht zugeschnitten. Die neu implementierten Sampler für Segregationsindikatoren erlauben, die Wahrscheinlichkeit zu berechnen, mit welcher ein bestimmtes QTL-Allel vererbt wird. Diese Information wird für die meisten zurzeit angewendeten Ansätze zur QTL-Kartierung benötigt. Darüber hinaus kann die-
se Wahrscheinlichkeit direkt dazu benutzt werden, um die Kovarianz-Matrix für Marker Assisted Best Linear Unbiased Prediction (BLUP) zu berechnen.
# Contents

1 Introduction 1
   1.1 Quantitative Trait Loci and Genetic Markers ............................... 1
   1.2 Linkage Analysis and QTL Mapping ........................................... 3
   1.3 Issues of QTL mapping .................................................................. 3
   1.4 Objectives of the Thesis ............................................................... 5

2 Theory 7
   2.1 Deterministic Maximum Likelihood ................................................. 7
      2.1.1 Elston-Stewart Algorithm ....................................................... 7
      2.1.2 Definitions ............................................................................ 12
      2.1.3 Application of the Elston-Stewart Algorithm ............................... 12
      2.1.4 Recursive Peeling ................................................................... 17
      2.1.5 Iterative Peeling ..................................................................... 19
      2.1.6 Algorithm to Approximate the Likelihood for Complex Pedigrees ................................. 31
      2.1.7 Summarizing ........................................................................... 33
   2.2 Markov Chain Monte Carlo Methods .............................................. 35
      2.2.1 Sampling Genotypes ................................................................ 36
      2.2.2 Sampling Blocks of Genotypes ............................................... 37
      2.2.3 ESIP Sampler ......................................................................... 38
      2.2.4 Sampling Allelic Origin ......................................................... 38
      2.2.5 Likelihood of Segregation Indicators ....................................... 44
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.2.6</td>
<td>Sampling Blocks of Segregation Indicators</td>
<td>47</td>
</tr>
<tr>
<td>2.2.7</td>
<td>Building Blocks randomly</td>
<td>51</td>
</tr>
<tr>
<td>2.2.8</td>
<td>More Loci</td>
<td>54</td>
</tr>
<tr>
<td>2.2.9</td>
<td>Haplotype Sampling</td>
<td>55</td>
</tr>
<tr>
<td>2.2.10</td>
<td>Cascading Origin Sampler</td>
<td>57</td>
</tr>
<tr>
<td>3</td>
<td>Combining Deterministic Calculation and Stochastic Simulation</td>
<td>61</td>
</tr>
<tr>
<td>3.1</td>
<td>MATVEC</td>
<td>61</td>
</tr>
<tr>
<td>3.1.1</td>
<td>Objects in MATVEC</td>
<td>62</td>
</tr>
<tr>
<td>3.2</td>
<td>Segregation Indicator Sampler</td>
<td>64</td>
</tr>
<tr>
<td>3.2.1</td>
<td>Setup</td>
<td>64</td>
</tr>
<tr>
<td>3.2.2</td>
<td>Initial Configuration</td>
<td>65</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Calculation of the Likelihood of a Descent Graph</td>
<td>66</td>
</tr>
<tr>
<td>3.2.4</td>
<td>The Segregation Indicator Sampler</td>
<td>66</td>
</tr>
<tr>
<td>3.3</td>
<td>Applications</td>
<td>69</td>
</tr>
<tr>
<td>3.4</td>
<td>Performance of Segregation Indicator Samplers</td>
<td>69</td>
</tr>
<tr>
<td>3.4.1</td>
<td>Single Nuclear Family</td>
<td>69</td>
</tr>
<tr>
<td>3.4.2</td>
<td>Half-sib Family</td>
<td>71</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Pedigree with Loops</td>
<td>75</td>
</tr>
<tr>
<td>3.4.4</td>
<td>Location Scores</td>
<td>77</td>
</tr>
<tr>
<td>3.4.5</td>
<td>More Loci</td>
<td>79</td>
</tr>
<tr>
<td>3.4.6</td>
<td>Large Pedigree</td>
<td>81</td>
</tr>
<tr>
<td>4</td>
<td>Discussion</td>
<td>85</td>
</tr>
<tr>
<td>4.1</td>
<td>Sampling Genotypes</td>
<td>85</td>
</tr>
<tr>
<td>4.2</td>
<td>Sampling Allelic Origin</td>
<td>86</td>
</tr>
</tbody>
</table>

Bibliography | 94 |
List of Figures

2.1 Simple pedigree .................................................. 12
2.2 Anterior members .................................................. 14
2.3 Posterior members ................................................ 14
2.4 Pedigree with four families ...................................... 22
2.5 Pedigree with loop ................................................. 23
2.6 Pedigree with loop ................................................. 25
2.7 Virtual pedigree, step a ........................................... 25
2.8 Pedigree with loop, step b ........................................ 25
2.9 Pedigree with loop, step c ........................................ 26
2.10 Cut-extended pedigree ............................................ 26
2.11 Cut-extended pedigree with coils ............................... 27
2.12 Reducible Gibbs sampler ........................................ 36
2.13 Non-communicating initial configurations .................... 37
2.14 Reducible Gibbs sampler ........................................ 37
2.15 Allelic origin ....................................................... 40
2.16 Segregation indicators .......................................... 40
2.17 Size of the state space: ordered genotypes .................. 41
2.18 Size of the state space: segregation indicators ............... 41
2.19 Connected components ......................................... 43
2.20 Non-communicating states: Descent graph A ................ 48
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.21</td>
<td>Non-communicating states: Descent graph C</td>
<td>48</td>
</tr>
<tr>
<td>2.22</td>
<td>Tunneling through incompatible states</td>
<td>50</td>
</tr>
<tr>
<td>2.23</td>
<td>$T_0$ transition rule</td>
<td>52</td>
</tr>
<tr>
<td>2.24</td>
<td>$T_1$ transition rule</td>
<td>52</td>
</tr>
<tr>
<td>2.25</td>
<td>$T_{2a}$ transition rule</td>
<td>53</td>
</tr>
<tr>
<td>2.26</td>
<td>$T_{2b}$ transition rule</td>
<td>53</td>
</tr>
<tr>
<td>2.27</td>
<td>Horizontal dependence</td>
<td>54</td>
</tr>
<tr>
<td>2.28</td>
<td>Haplotype sampler failing in horizontal dependence</td>
<td>58</td>
</tr>
<tr>
<td>3.1</td>
<td>Nuclear family</td>
<td>70</td>
</tr>
<tr>
<td>3.2</td>
<td>Nuclear family in segregation indicator representation</td>
<td>70</td>
</tr>
<tr>
<td>3.3</td>
<td>Half-sib family</td>
<td>73</td>
</tr>
<tr>
<td>3.4</td>
<td>Half-sib family in segregation indicator representation</td>
<td>73</td>
</tr>
<tr>
<td>3.5</td>
<td>Half-sib family in segregation indicator representation</td>
<td>73</td>
</tr>
<tr>
<td>3.6</td>
<td>Half-sib family in segregation indicator representation</td>
<td>73</td>
</tr>
<tr>
<td>3.7</td>
<td>Half-sib family in segregation indicator representation</td>
<td>74</td>
</tr>
<tr>
<td>3.8</td>
<td>Test pedigree with ABO-blood-type data</td>
<td>75</td>
</tr>
<tr>
<td>3.9</td>
<td>Artificial pedigree with loops</td>
<td>78</td>
</tr>
<tr>
<td>3.10</td>
<td>Grid of QTL location scores</td>
<td>78</td>
</tr>
<tr>
<td>3.11</td>
<td>Pedigree with 2 markers</td>
<td>79</td>
</tr>
<tr>
<td>3.12</td>
<td>Pedigree with 2 markers and a recombinant offspring</td>
<td>80</td>
</tr>
<tr>
<td>3.13</td>
<td>Complex pedigree with 325 members</td>
<td>82</td>
</tr>
<tr>
<td>4.1</td>
<td>Simple pedigree demonstrating vertical dependence</td>
<td>88</td>
</tr>
<tr>
<td>4.2</td>
<td>Vertical dependence overcome by blocking</td>
<td>89</td>
</tr>
<tr>
<td>4.3</td>
<td>Inefficient blocking</td>
<td>90</td>
</tr>
</tbody>
</table>
# List of Tables

3.1 Genotype probabilities in nuclear family ............................................ 71
3.2 Genotype probabilities in half-sib family ........................................... 72
3.3 Genotype probabilities in pedigree with loops ..................................... 76
3.4 Genotype probabilities in pedigree with loops ..................................... 77
3.5 Proportion of ordered genotypes ....................................................... 80
3.6 Proportion of ordered genotypes ....................................................... 81
3.7 PDQ's of male offspring in generation 4 ............................................. 83
Chapter 1

Introduction

Knowledge about the presence, the mode of inheritance and the chromosomal location of genes is essential to gain a better understanding of transmission of diseases and performance traits.

Genetic data can be used to test for the presence of genes and to estimate the location and effects of these genes.

1.1 Quantitative Trait Loci and Genetic Markers

The hereditary units in an individuals genome are defined as genes. A gene can be considered as a functional entity in the DNA sequence encoding a specific protein or having some regulatory function within the DNA metabolism. At the molecular level, genes have a complex structure with introns, exons and regulatory sequences.

The position of a gene is called a *locus* and refers to a single location on the chromosome, a gene may extend over several thousands of base-pairs. A gene is said to be polymorphic if it has at least two distinct variants which cause a measurable effect in the phenotypic expression of a trait. Theses variants are called *alleles*.
The genetic principles which have first been proposed by Mendel were based on the observation of qualitative differences like coat or eye color which divide individuals into sharply distinct types. Theses readily detectable differences allowed to derive Mendelian segregation ratios reflecting the underlying mechanism of inheritance.

Many important functions and traits do not show such sharp differentiation in kind or quality. Their more or less pronounced variations in size or quantity do not fall in demarcated classes, they rather form continuously graded series from one extreme to the other. Many aspects of evolution and almost all characters of economic importance in animal breeding involve quantitative traits. Quantitative differences are most often influenced by several genes with relatively small effects and considerable environmental influences. The intrinsically discontinuous variations caused by any single gene is not large enough to cause a recognizable discontinuity. Continuous variation added through non-genetic sources may further hide the discontinuous nature of the underlying gene effects. Thus, in general, direct observation of those gene effects is not possible in general. Although these genes cannot be identified by “simple Mendelian” analysis, they do follow the Mendelian laws of inheritance. These genes are called quantitative trait loci (QTL, Gelderman, 1975).

As the effect and the location of a QTL is a priori unknown, their direct observation is not possible. This constraint can be circumvented by taking into account information about position and inheritance of known and precisely located loci on the genome and modeling co-segregation of a putative QTL with such fix-points. These fix-points are usually called markers. They are DNA sequences which exhibit any form of variability that can be detected by some means. Although markers are not necessarily genes as defined above the different variants of a marker are also called alleles and its position on the genome is again called locus.

With the development of molecular genetics, markers can be defined on the DNA level, i.e. as variants of the DNA sequence. Such markers may be restriction fragment length polymorphisms (RFLP) or the more recently detected micro-satellites and single nucleotide polymorphisms (SNP).
1.2 Linkage Analysis and QTL Mapping

The statistical analysis of a putative co-segregation of a marker and a QTL is called *linkage analysis*. Loci are referred to as *linked*, if their alleles do not segregate independently of each other. If co-segregation (linkage) of a marker locus and a putative QTL is present, the relative position of the QTL can be deducted from the known position of the marker locus. Such information may subsequently be used to position a QTL within a marker map (*QTL mapping*). The use of many markers as references in a *multi-point* linkage analysis further enhances the accuracy of QTL position estimates. In contrast to the common *two-point* linkage analysis where the QTL position is estimated relative to a single known marker, *multi-point* linkage analysis allows the detection of double recombinations and provides information in cases where not all meiosis are informative. For the efficient sequencing of QTLs, precise mapping of their position is a prerequisite.

Although today accurate and dense marker maps for detailed screening of the genome are as readily available as data on economically important traits in livestock, appropriate statistical models to screen arbitrary pedigrees for QTL effects and positions do not exist and are thus needed.

1.3 Issues of QTL mapping

In traditional quantitative genetics, the *infinitesimal model* (Bulmer, 1980) has been used successfully for years to deal with quantitative traits in livestock. It assumes an infinite number of loci each having infinitesimally small effect on a trait. The effects of the QTL are considered jointly forming the additive and dominance genetic variance for a certain trait, which allows efficient calculations (Henderson, 1973; Henderson, 1984; Wiggans et al., 1988). However, this model by design reveals no indication about linkage.

With the development of an efficient algorithm to compute the likelihood of large sim-
ple pedigrees for monogenetic and oligogenetic inheritance by Elston and Stewart (1971), QTL mapping became possible for non-complex pedigrees and monogenetic or oligogenetic inheritance. Their approach, often referred to as peeling has been successfully applied in humans to elucidate the inheritance of Mendelian disorders. However, there are three major issues that prevent the direct application of the Elston-Stuart algorithm for QTL mapping:

1. The Complexity of the Pedigree Structure
   The Elston-Stewart algorithm fails in complex pedigrees, i.e. pedigrees with inbreeding or marriage loops. Although extensions to account for loops have been proposed for small pedigrees (Lange and Elston, 1975; Cannings et al., 1978; Lange and Boehnke, 1983), these approaches are not suitable for the size and complexity of general livestock pedigrees. Wang et al. (1996) proposed an approximation to the likelihood for oligogenic models in large and complex pedigrees with loops based on the original work of Elston and Stewart (1971) and the idea of iterative peeling by Janss et al. (1992).

   Markov Chain Monte Carlo (MCMC) methods have been put forward (Thompson and Wijsman, 1990; Thompson and Guo, 1991; Thompson, 1991; Thomas and Cortes-sis, 1992; Janss et al., 1994) to analyze complex pedigrees. MCMC can efficiently account for arbitrary pedigree structures as long as the sampling procedure ensures that the resulting Markov chain is aperiodic and irreducible, i.e. that all possible states within the state space may be sampled with probability $> 0$.

2. The Number of Loci in the Model
   Calculating the likelihood for an oligogenetic model requires to sum over unobserved phase known genotypes, i.e. to consider all haplotype combinations a parent possibly transmitted to the offspring. Therefore, models accounting for more than a few loci soon become computationally infeasible.

   Mixed inheritance, i.e. few major loci with large effect and many minor loci with
small effects also invalidate deterministic likelihood calculation, when the effect of the
minor loci is assumed to be normally distributed (Elston and Stewart, 1971; Morton
and MacLean, 1974), and thus approximations have to be used (Hasstedt, 1982;
familial correlations. In extended pedigrees, their approach is only approximate and
becomes inefficient when phenotypic observations are missing. Fernando et al. (1994)
and Strieker et al. (1995b) proposed an approximation to mixed inheritance with a
mixture of discrete distributions originating from the segregation of a finite number
of genetic loci.
MCMC on the other hand is completely general with respect to the genetic model
assumed. Realizations based on any number of QTL and any distribution assumed
can be simulated by an appropriate sampler. However, large numbers of loci in the
model may reveal huge state spaces to simulate from.

3. Missing Data
If only a small portion of the genotypes at many loci are observed, the number of
summations in a deterministic approach and the state space in a MCMC-approach in¬
crease. This will reveal deterministic likelihood calculations and MCMC simulations
increasingly inefficient. In this respect the missing data problem and the number of
loci considered are connected.

1.4 Objectives of the Thesis

MCMC-based approaches usually sample the set of genetic descent states of a pedigree
where each state specifies the paths of gene flow and the alleles inherited by each path.
Recently, Thompson (1994) showed that linkage information is in the origin of an allele
and not in the specific allele transmitted. All allelic origins in a pedigree are also referred
to as transmission patterns or genetic descent graph and consist of a set of segregation in¬


indicators describing the origin of alleles in a pedigree. This leads to a significant reduction of the state space to simulate from. Sobel and Lange (1996) extended this approach by allowing multiple updating steps and applying a Metropolis-Hastings sampler (Metropolis et al., 1953; Hastings, 1970) to ensure irreducible Markov chains.

The objective of this thesis is to use this new stochastic approach together with deterministic maximum likelihood for efficient QTL-mapping. The basic idea is to retain deterministic Maximum Likelihood at a biallelic QTL while the inheritance at multiple markers is considered by different Markov chain Monte Carlo approaches. Since deterministic ML is retained for inference about the QTL, the problem of complex pedigrees (issue 1 above) will by addressed by the approach of Wang et al. (1996). All MCMC approaches considered will perform simulations on the state space of transmission patterns.
Chapter 2

Theory

In this chapter, available methods for linkage analysis using deterministic and stochastic methods are reviewed.

2.1 Deterministic Maximum Likelihood

2.1.1 Elston-Stewart Algorithm

Given some model of inheritance, the probability or the probability density of the phenotypic values of the pedigree members, expressed as a function of the unknown parameters, is the likelihood of the model. For simplicity, the distinction between a probability and a probability density will be omitted from hereon.

The efficient algorithm to compute the likelihood of simple pedigrees for monogenetic and oligogenetic inheritance given by Elston and Stewart (1971) set the stage of modern linkage analysis. This algorithm is often referred to as peeling. In a pedigree, i.e. a set of related individuals, phenotypic values are usually associated with each member, but it is possible for some members to have missing genotype information. Thus, a pedigree contains infor-
mation on the relationships between its members and about a trait of interest.

Under oligogenic inheritance, phenotypic values are assumed to be conditionally independent given the genotypes and it is assumed that the marker genotype has no effect upon the phenotype. Assuming that markers have no influence on the observed quantitative trait, the conditional probability of the phenotypic values given \( u \), the vector of joint genotypes at a postulated quantitative trait locus (QTL) and the marker loci, for \( n \) pedigree members can thus be written as

\[
Pr(y | u) = Pr(y | g) = \prod_{i=1}^{n} Pr(y_i | g_i),
\]

where \( y \) is the vector of \( n \) phenotypic values, \( g \) is the vector of \( n \) genotypes at the QTL, and \( Pr(y_i | g_i) \) is the penetrance function or the conditional probability of the phenotypic value of individual \( i \) given the genotype \( g_i \) at the QTL. Under Mendelian inheritance, the probability of the joint genotypes \( u \) at the QTL and the marker loci can be written as

\[
Pr(u) = \prod_{i=1}^{n_1} Pr(u_i) \prod_{i=n_1+1}^{n} Pr(u_i | u_m, u_f),
\]

where pedigree members 1 through \( n_1 \) are founders and the rest are non-founders. \( Pr(u_i) \) is the population frequency of the joint genotype at the QTL and the marker loci. \( Pr(u_i | u_m, u_f) \) is the transition probability or the conditional probability that an offspring will have the joint genotype \( u_i \) given the mother \( m \) has the joint genotype \( u_m \) and the father \( f \) has the joint genotype \( u_f \). Under the assumption of one QTL and one marker, let \( T_b, T_d \) be alleles at the QTL and \( M_c, M_e \) be alleles at the marker locus. \( Pr(T_b T_d) \) is the marginal probability of the genotype at the QTL in the population and \( Pr(M_c M_e) \) is the marginal probability of the marker genotype in the population. Then, \( Pr(u_i) \), the probability of the joint genotype in the population, can be computed as (Elston and Stewart, 1971)
2.1 Deterministic Maximum Likelihood

\[ \Pr(u_i) = \Pr \left[ \frac{T_b M_c}{T_d M_e} \right] = C \cdot \Pr(T_b T_d) \cdot \Pr(M_c M_e), \]

where

\[ C = \begin{cases} 1 & \text{if } T_b = T_d \text{ or } M_c = M_e, \\ \frac{1}{2} & \text{otherwise.} \end{cases} \] (2.3)

To compute the transition probability \( \Pr(u_i \mid u_m, u_f) \), i.e. the conditional probability that an offspring will have the joint genotype \( u_i \), given its parents have the joint genotypes \( u_m \) and \( u_f \), respectively, we first define the transmission probabilities \( \tau_s \left[ \frac{T_b M_c}{T_d M_e} \rightarrow T_f M_g \right] \) following Elston and Stewart (1971): \( \tau_s \left[ \frac{T_b M_c}{T_d M_e} \rightarrow T_f M_g \right] \) is the probability that a parent of sex \( s \) with joint genotype \( \left[ \frac{T_b M_c}{T_d M_e} \right] \) will transmit the haplotype \( [T_f M_g] \) to the offspring. These transmission probabilities are computed as:

\[ \tau_s \left[ \frac{T_b M_c}{T_d M_e} \rightarrow T_f M_g \right] = \frac{(1 - \theta_s) (\delta_{T_b T_f} \delta_{M_c M_g} + \delta_{T_d T_f} \delta_{M_c M_g})}{2} + \frac{\theta_s (\delta_{T_b T_f} \delta_{M_c M_g} + \delta_{T_d T_f} \delta_{M_c M_g})}{2}, \] (2.4)

where \( \theta_s \) is the sex dependent recombination fraction between the trait and the marker locus and \( \delta_{xy} \) equals 1 if \( x = y \), 0 otherwise. Using these transmission probabilities, the transition probability \( \Pr(u_i \mid u_m, u_f) \) can be calculated as:
Pr(u_i | u_m, u_f) = Pr \left[ \frac{T_f M_g}{T_h M_j}, \frac{T_i M_s}{T_a M_v}, \frac{T_b M_c}{T_d M_e} \right]

= \begin{cases} 
\tau_m \left[ \frac{T_i M_s}{T_a M_v} \rightarrow T_h M_j \right] \tau_f \left[ \frac{T_b M_c}{T_d M_e} \rightarrow T_f M_g \right] & \text{if } T_f = T_h \text{ and } M_g = M_j, \\
\tau_m \left[ \frac{T_i M_s}{T_a M_v} \rightarrow T_f M_j \right] \tau_f \left[ \frac{T_b M_c}{T_d M_e} \rightarrow T_f M_g \right] + \\
\tau_m \left[ \frac{T_i M_s}{T_a M_v} \rightarrow T_f M_g \right] \tau_f \left[ \frac{T_b M_c}{T_d M_e} \rightarrow T_h M_j \right] & \text{otherwise.}
\end{cases}

(2.5)

The likelihood for the pedigree can then be computed as

$$L(y) = \sum_{u_1} \sum_{u_2} \cdots \sum_{u_n} \prod_{i=1}^{n} Pr(y_i | g_i) \prod_{i=1}^{n_1} Pr(u_i) \prod_{i=n_1+1}^{n} Pr(u_i | u_m, u_f).$$

(2.6)

The summations in (2.6) have to be carried out for all joint genotypes. For founders, let

$$f(u_i) = Pr(y_i | g_i) Pr(u_i)$$

and for non-founders, let

$$h(u_i, u_m, u_f) = Pr(y_i | g_i) Pr(u_i | u_m, u_f)$$

Then, the likelihood can be written as
2.1 Deterministic Maximum Likelihood

\[ L(y) = \sum_{u_1} \sum_{u_2} \cdots \sum_{u_n} \prod_{i=1}^{n_1} f(u_i) \prod_{i=n_1+1}^{n} h(u_i, u_m, u_f). \] (2.7)

If two alleles at each of the QTL and the marker locus are assumed, 10 joint genotypes at the QTL and the marker locus have to be considered, because the doubly heterozygous genotype exists in two phases. Thus, each summation in (2.7) would be over these 10 joint genotypes. If the marker genotypes can be observed for all \( n \) pedigree members, then the likelihood can be written as

\[ L(y) = \sum_{g_1} \sum_{g_2} \cdots \sum_{g_n} \prod_{i=1}^{n_1} f(u_i) \prod_{i=n_1+1}^{n} h(u_i, u_m, u_f), \] (2.8)

i.e. only the summations over the genotypes \( g \) at the QTL have to be carried out. However, different phases at the QTL of double or multiple heterozygous individuals need to be considered. If the summations are over \( m \) joint genotypes, the number of calculations required to compute the likelihood as indicated by (2.7) is proportional to \( m^n \). Lange and Boehnke (1983) among others showed how the amount of necessary summations may be reduced using genotype and phase elimination. Furthermore, because the function \( f(u_i) \) involves the joint genotype of only a founder and the function \( h(u_i, u_m, u_f) \) involves the joint genotypes of only a non-founder and parents \( m \) and \( f \), the order of adding and multiplying in (2.7) can be rearranged such that the number of calculations required to compute the likelihood is proportional to \( n \) (Elston and Stewart, 1971; Cannings et al., 1978). Such rearrangement of the summations will be demonstrated in section 2.1.3.
2.1.2 Definitions

The pedigree in Figure 2.1 consists of two families which are connected by the connector individual 5. A connector belongs always at least to two families. A nuclear family is defined as one family consisting of a mother and a father and their common offspring. A family with only a single connector is defined to be a terminal family; both families in Fig. 2.1 are terminal families.

The individuals 1, 2 and 3 are referred to as founders as neither of their parents are included in the pedigree. The individuals 4, 6 and 7 are terminal members as they do not have any offspring included in this pedigree.

The family in which an individual is an offspring is defined as the individual’s anterior family, and the family in which an individual is a parent is defined as the posterior family of the individual through the other parent, so the posterior family of individual 5 through its mate 3 consists of these parents and the offspring 6 and 7. Any individual has at most one anterior family (none for founders) and has, except for terminal members, posterior families through each of its mates.

2.1.3 Application of the Elston-Stewart Algorithm

For the pedigree in Figure 2.1 with the founder individuals 1, 2, 3 and the offspring 4, 5, 6 and 7, the likelihood for an oligogenic model is (assuming again that markers do not


2.1 Deterministic Maximum Likelihood

influence the quantitative trait):

\[
L(y) = \sum_{u_1} \sum_{u_2} \sum_{u_3} \sum_{u_4} \sum_{u_5} \sum_{u_6} \sum_{u_7} \prod_{i=1}^{7} \Pr(y_i \mid g_i) \prod_{i=1}^{3} \Pr(u_i) \prod_{i=4}^{7} \Pr(u_i \mid u_m, u_f)
\]

\[
= \sum_{u_1} \sum_{u_2} \sum_{u_3} \sum_{u_4} \sum_{u_5} \sum_{u_6} \sum_{u_7} \Pr(y_1 \mid g_1) \Pr(y_2 \mid g_2) \Pr(y_3 \mid g_3) \Pr(y_4 \mid g_4) \Pr(y_5 \mid g_5) \Pr(y_6 \mid g_6) \Pr(y_7 \mid g_7)
\]

\[
\times \Pr(u_1) \Pr(u_2) \Pr(u_3)
\]

\[
\times \Pr(u_4 \mid u_1, u_2) \Pr(u_5 \mid u_1, u_2) \Pr(u_6 \mid u_5, u_3) \Pr(u_7 \mid u_5, u_3)
\]

(2.9)

Now, the summations can be rearranged according to the genotypes they depend on:

\[
L(y) = \sum_{u_1} \Pr(y_1 \mid g_1) \Pr(u_1)
\]

\[
\sum_{u_2} \Pr(y_2 \mid g_2) \Pr(u_2)
\]

\[
\sum_{u_4} \Pr(y_4 \mid g_4) \Pr(u_4 \mid u_2, u_1)
\]

\[
\sum_{u_5} \Pr(y_5 \mid g_5) \Pr(u_5 \mid u_2, u_1)
\]

(2.10)

\[
\sum_{u_3} \Pr(y_3 \mid g_3) \Pr(u_3)
\]

\[
\sum_{u_6} \Pr(y_6 \mid g_6) \Pr(u_6 \mid u_5, u_3)
\]

\[
\sum_{u_7} \Pr(y_7 \mid g_7) \Pr(u_7 \mid u_5, u_3)
\]

Compared to equation 2.9, this reduces the number of summations that need to be carried out considerably.

Relative to any member \(i\) in a pedigree without loops, the remaining members can be divided into two groups, those anterior to it and those posterior to it. The members
anterior to any member \( i \) are connected to \( i \) through its parents and full sibs including the parents and full sibs themselves. In the example above, the members anterior to individual 5 are 1, 2 and 4 (figure 2.2).

The pedigree members posterior to an individual \( i \) are connected to \( i \) through its mates and offspring including the mates and offspring themselves. The members posterior to individual 5 are 3, 6 and 7 (Fig. 2.3).

This property can now be used to put together or peel all information anterior to individual 5 and the genotype of individual 5 by calculating (c.f. 2.10, Fig. 2.2)
\[ \Pr(y_1, y_2, y_4, u_5) = a_5(u_5) = \sum_{u_1} Pr(y_1 \mid g_1) \Pr(u_1) \sum_{u_2} Pr(y_2 \mid g_2) \Pr(u_2) \]
\[ \times \left[ \sum_{u_4} Pr(y_4 \mid g_4) \Pr(u_4 \mid u_1, u_2) \right] \]
\[ \times [\Pr(u_5 \mid u_1, u_2)] , \]  

This is called the \textit{anterior value} of individual 5 as anterior probabilities do not sum to unity. The anterior value of individual 5 is proportional to the joint probability of the pedigree members \textit{anterior} to individual 5 and its genotype \( u_5 \).

Now, this anterior value \( a_5(u_5) \) of individual 5 can be used in (2.10) to get

\[ \Pr(y) = \sum_{u_5} Pr(y_5 \mid g_5) \Pr(y_1, y_2, y_4, u_5) \]
\[ \times \sum_{u_3} Pr(y_3 \mid g_3) \Pr(u_3) \]
\[ \times \left[ \sum_{u_6} Pr(y_6 \mid g_6) \Pr(u_6 \mid u_5, u_3) \right] \]
\[ \times \left[ \sum_{u_7} Pr(y_7 \mid g_7) \Pr(u_7 \mid u_5, u_3) \right] \]
\[ = \sum_{u_5} Pr(y_5 \mid g_5) a_5(u_5) \]
\[ \times \sum_{u_3} Pr(y_3 \mid g_3) \Pr(u_3) \]
\[ \times \left[ \sum_{u_6} Pr(y_6 \mid g_6) \Pr(u_6 \mid u_5, u_3) \right] \]
\[ \times \left[ \sum_{u_7} Pr(y_7 \mid g_7) \Pr(u_7 \mid u_5, u_3) \right] \]

(2.12)
Collapsing anterior information onto an individual $i$ as in (2.12) is referred to as peeling down.

Alternatively, all information posterior to an individual $i$ through a specific mate $j$ of $i$ can be collapsed onto $i$ (peeling up, Cannings et al. (1978); Fig. 2.3) by

\[
\Pr(y_3, y_6, y_7 \mid u_5) = p_{5,3}(u_5) = \sum_{u_3} Pr(y_3 \mid g_3) Pr(u_3) \times \sum_{u_6} Pr(y_6 \mid g_6) Pr(u_6 \mid u_5, u_3) \times \sum_{u_7} Pr(y_7 \mid g_7) Pr(u_7 \mid u_5, u_3)
\]

(2.13)

The posterior value $p_{ij}(u_i)$ of $i$ through mate $j$, $p_{ij}(u_i)$ is proportional to the conditional probability of the phenotypes of the pedigree members posterior to $i$ through its mate $j$ and their common offspring, given $i$ has genotype $u_i$. This posterior value and the above anterior value can now be used to calculate the likelihood

\[
L(y) = \sum_{u_1} Pr(y_1 \mid g_1) Pr(u_1) \sum_{u_2} Pr(y_2 \mid g_2) Pr(u_2) \sum_{u_4} Pr(y_4 \mid g_4) Pr(u_4 \mid u_1, u_2) \sum_{u_5} Pr(y_5 \mid g_5) Pr(u_5 \mid u_1, u_2) Pr(y_3, y_6, y_7 \mid u_5) = \sum_{u_5} a_5(u_5) Pr(y_5 \mid g_5)p_{5,3}(u_5)
\]

(2.14)
2.1 Deterministic Maximum Likelihood

2.1.4 Recursive Peeling

In general, the likelihood for a pedigree without loops can be computed recursively through any member \( i \) by (Fernando et al., 1993):

\[
L(y) = \sum_{u_i} a_i(u_i) g(y_i | g_i) \prod_{j \in S_i} p_{ij}(u_i) \tag{2.15}
\]

Again, \( y \) is the vector of phenotypes \( (y_i) \), \( u_i \) is the genotype of individual \( i \), \( a_i(u_i) \) the anterior value, \( g(y_i | g_i) \) the penetrance value and \( p_{ij}(u_i) \) is the posterior value through mate \( j \) within \( S_i \), the set of mates of \( i \).

To calculate the likelihood recursively starting from an arbitrary individual \( i \), the anterior value \( a_i(u_i) \) and the posterior value \( p_{ij}(u_i) \) for \( j \in S_i \) are needed. To compute the anterior value for \( i \), it is required to compute anterior and posterior values for each of its parents and posterior value for each of its full sibs. To compute the posterior value for individual \( i \) through its mate \( j \), it is required to calculate again anterior and posterior values for mate \( j \) and posterior values for their common offspring.

As founder individuals do not have anterior members, their anterior values are given by the appropriate population genotype probabilities. If \( i \) is not a founder, it is connected to its anterior members only through its parents and full sibs. Thus, to recursively calculate the anterior probabilities for \( i \), the following is to be calculated:

1. anterior probabilities \( a_m(u_m) \) and \( a_f(u_f) \) for mother \( m \) and father \( f \) of \( i \).
2. posterior probabilities \( p_{mj}(u_m) \) for \( m \) through all their mates \( j \) except \( f \).
3. posterior probabilities \( p_{fj}(u_f) \) for \( m \) through all his mates \( j \) except \( m \).
4. the full sibs posterior probabilities \( p_{jk}(u_j) \) for all offspring \( j \) of \( m \) and \( f \) except \( i \) through their mates.
Once these are calculated, $a_i(u_i)$ can be calculated as

$$a_i(u_i) = \sum_{u_m} a_m(u_m) \Pr(y_m \mid g_m) \prod_{j \in S_m, j \neq i} p_{mj}(u_m) \sum_{u_f} a_f(u_f) \Pr(y_f \mid g_f) \prod_{j \in S_f, j \neq m} p_{fj}(u_f) \Pr(u_i \mid u_m, u_f)$$

$$\prod_{j \in C_{mf}, j \neq i} \left[ \sum_{u_j} \Pr(u_j \mid u_m, u_f) \Pr(y_j \mid g_j) \prod_{k \in S_j} p_{jk}(u_j) \right]$$

(2.16)

where $C_{mf}$ is the set of children of the parents $m$ and $f$.

Pedigree member $i$ is connected to its posterior members only through its mates and offspring. Thus, to recursively calculate the posterior values $p_{ij}(u_i)$ for $i$ through its mate $j$, the following is to be calculated:

1. anterior values $a_j(u_j)$ for mate $j$ of $i$

2. posterior values $p_{jk}(u_j)$ for $j$ through all the mates $k$ of $j$ except $i$

3. posterior values $p_{oi}(u_o)$ for all offspring $o$ of $i$ and $j$ through their mates $l$

Once these are available, $p_{ij}(u_i)$ can be calculated as

$$p_{ij}(u_i) = \sum_{u_j} a_j(u_j) \Pr(y_j \mid g_j) \prod_{o \in S_j, j \neq i} p_{jo}(u_j) \times \prod_{o \in C_{ij}} \left[ \sum_{u_o} \Pr(u_o \mid u_i, u_j) \Pr(y_o \mid g_o) \prod_{l \in S_o} p_{ol}(u_o) \right]$$

(2.17)
2.1 Deterministic Maximum Likelihood

This implicitly collapses the pedigree information to one single nuclear family. If this reduction cannot be done, i.e., the pedigree can not be divided into an anterior and a posterior part relative to any arbitrary individual, the pedigree contains inbreeding or marriage loops. In this case, the recursive peeling algorithm will fail as anterior and posterior values of an individual are no longer properly defined.

2.1.5 Iterative Peeling

Complex pedigrees with inbreeding and/or marriage loops are very common in livestock breeding. In small pedigrees frequently encountered in human genetics, this problem may be solved by loop cutting. Each loop in the pedigree has to be identified and subsequently cut by introducing an additional founder into the pedigree which creates a modified pedigree without loops. The computation of the exact likelihood for the original pedigree is possible by a modified Elston-Stewart algorithm (Lange and Elston, 1975). Cannings et al. (1978) described a recursive algorithm to compute the exact likelihood for pedigree with loops. Both algorithms require the identification of all loops in a pedigree and are thus restricted to relatively small pedigrees that allow visual inspection of their structure.

Lange and Boehnke (1983) and Strieker et al. (1995a) demonstrated methods that do not require the identification of the loops in a pedigree. Unfortunately, the application of the first is again restricted to small pedigrees due to exponential relation between number of loops and computing workload and/or storage requirements while the second approach is only an approximation.

Alternatively, Janss et al. (1992) proposed an iterative algorithm to calculate the likelihood in arbitrary pedigrees. Wang et al. (1996) showed that this iterative approach does implicitly cut loops and extend the pedigree by introducing artificial individuals and presented a modified version of the iterative algorithm using the concept of anterior and posterior values by Fernando et al. (1993). Subsequently, iterative peeling for simple and complex pedigrees and the modifications of this approach by Wang et al. (1996) will be explained.
Iterative Peeling in Pedigrees without Loops

In contrast to recursive peeling starting recursion at an arbitrary pedigree member, iterative peeling involves for all pedigree members:

1. for each pedigree member \( i \) to initialize anterior values to the population genotype frequency and posterior values to unity subsequently calculate its penetrance values.

2. Then, for each connector \( i \)

   - for families in which \( i \) is an offspring, its anterior value \( a_i(u_i) \) is calculated non-recursively using (2.16) and the current values of the required quantities.
   - for families in which \( i \) is a parent, its posterior value \( p_{ij}(u_i) \) through each mate \( j \) is calculated non-recursively using (2.17) and the current values of the required quantities.

Additionally, anterior, posterior and penetrance values may be scaled to sum to unity over genotype as described in Wang et al. (1996) to avoid numerical underflow. The log scaling factors \( K_{a_i}, K_{p_i} \) and \( K_{p_{ij}} \) for the anterior, posterior and penetrance values are also calculated using the current values of the required quantities. They denote the accumulative log of the scaling factors for the anterior, posterior and penetrance values for member \( i \) with its mother \( m_i \) and father \( p_i \). These three log scaling factors are computed as:
2.1 Deterministic Maximum Likelihood

\[ K_{a_i} = K_{a_{m_i}} + K_{g_{m_i}} + \sum_{j \in S_{m_i}, j \neq i} K_{p_{m_i,j}} + K_{a_{f_i}} + K_{g_{f_i}} + \sum_{j \in S_{f_i}, j \neq m_i} K_{p_{f_i,j}} \]

\[ + \sum_{j \in C_{m_i}, j \neq i} \left( K_{g_j} + \sum_{k \in S_j} K_{p_{j,k}} \right) + \log \left[ \sum_{u_i} a_i(u_i) \right] \]

\[ K_{p_{ij}} = K_{a_j} + K_{g_j} + \sum_{k \in S_j, k \neq i} K_{p_{j,k}} + \sum_{k \in C_{ij}} \left( K_{g_k} + \sum_{l \in S_k} K_{p_{k,l}} \right) + \log \left[ \sum_{u_i} p_{ij}(u_i) \right] \]

\[ K_{g_i} = \log \left[ \sum_{u_i} g(y_i | u_i) \right] \]

(2.18)

3. Step 2 is repeated until each of the anterior and posterior values have converged.

Convergence is usually achieved in less than 10 iterations.

The likelihood for a pedigree without loops can now be computed through any connector \( i \):

\[ \log L(y) = \log \left[ \sum_{u_i} \hat{a}_i(u_i) \hat{y}(y_i | u_i) \prod_{j \in S_i} \hat{p}_{ij}(u_i) \right] \]

\[ + K_{a_i} + K_{g_i} + \sum_{j \in S_i} K_{p_{ij}} \]

(2.19)

where \( \hat{a}_i(u_i) \), \( \hat{y}(y_i | u_i) \) and \( \hat{p}_{ij}(u_i) \) are the scaled values of \( a_i(u_i) \), \( g(y_i | u_i) \) and \( p_{ij}(u_i) \).

**Sequence Optimization:** Although convergence in iterative peeling does not depend on the sequence of computations, efficiency may vary between different sequences. Wang et al. (1996) showed that it is possible to find for pedigree without loops a sequence of cal-
calculations to compute the anterior and posterior values such that convergence is achieved in one iteration. Such a peeling sequence will be referred to as an optimal peeling sequence (OPS). To illustrate consider the pedigree in Fig. 2.4:

![Pedigree Diagram](image)

*Figure 2.4: Pedigree consisting of four families with connectors A, B and C*

If iterations are performed over connectors in the sequence \([B - A - C]\), then

in the first iteration

- (b) \([a_B(u_B), p_{B,A}(u_B), p_{B,3}(u_B)]\) are calculated with \(p_{B,A}(u_B)\) and \(p_{B,3}(u_B)\) not converged,
- (a) \([a_A(u_A), p_{A,B}(u_A)]\), are calculated with \(p_{A,B}(u_A)\) not converged and
- (c) \([a_C(u_C), p_{C,5}(u_C)]\), are calculated with \(a_C(u_C)\) not converged.

In the second iteration, all anterior and posterior values will converge:

- (b) \([a_B(u_B), p_{B,A}(u_B), p_{B,3}(u_B)]\) as \(a_A(u_A)\) and \(p_{C,5}(u_C)\) have been computed in iteration 1,
- (a) \([a_A(u_A), p_{A,B}(u_A)]\) as \(p_{B,3}(u_B)\) has been computed and
- (c) \([a_C(u_C), p_{C,5}(u_C)]\) as \(a_C(u_C)\) has been computed.
2.1 Deterministic Maximum Likelihood

If an optimum peeling sequence (OPS) like \([A - C - B]\) is applied, then convergence is achieved in one iteration:

first iteration:

(a) \([a_A(u_A), p_{A,B}(u_A)]\) \((p_{A,B}(u_A)\) not converged),

(b) \([a_C(u_C), p_{C,5}(u_C)]\) \((a_C(u_C)\) not converged) and

(c) \([a_B(u_B), p_{B,A}(u_B), p_{B,3}(u_B)]\); in this step \(p_{B,A}(u_B)\) and \(p_{B,3}(u_B)\) are converged.

The likelihood can now be computed as:

\[
L(y) = \sum_{u_B} a_B(u_B) \cdot g(y_B | u_B) \cdot p_{B,A}(u_B) \cdot p_{B,3}(u_3)
\]  

Finding Optimum Peeling Sequences  To determine an OPS and calculate the likelihood, a pedigree is peeled onto one single family (terminal peeling) by clipping away step by step all terminal families, i.e. families which have only one connector (Wang et al., 1996). For this purpose, an arbitrary terminal family is determined and its anterior

![Figure 2.5: Optimum peeling sequence: Peeling away of terminal families](image)

value is computed if the connector is an offspring or its posterior value through the other
parent in the family is computed respectively if the connector is a parent. In a second step, all non-connector members of the selected terminal family are removed from the pedigree. These two steps are repeated until only one family remains in the pedigree.

To illustrate, consider again the pedigree in Fig. 2.4: Choosing first the terminal family (1, 2; A), the anterior value $a_A(u_A)$ is calculated as A is offspring in this family, then 1 and 2 are removed (Fig. 2.5: I). Then, choosing family (C, 5; 6), $p_{C,5}(u_C)$ is calculated and 5 and 6 removed from the pedigree (Fig. 2.5: II). Now, the pedigree consists of two families, (A, B; 4) and (B, 3; C), which are now themselves terminal families as shown on the right side of Fig. 2.5. By choosing family (A, B; 4), calculating $p_{B,A}(u_B)$ and removing A and 4 (Fig. 2.5: III), the pedigree is reduced to a nuclear family. Note that whenever a OPS is executed, all quantities required to compute anterior or posterior values in a specific step will have been computed in previous steps, e.g. the calculation of $p_{B,A}(u_B)$ requires $a_A(u_A)$, which has been computed in the first step. The likelihood can now be computed by this OPS $[a_A(u_A), p_{C,5}(u_C), p_{B,A}(u_B)]$ using (2.19). Note that an OPS is not unique, $[a_A(u_A), p_{B,A}(u_B), a_C(u_C)]$ is also an OPS.

Terminal peeling following an OPS in a pedigree without loops thus avoids repeated computations and reveals iterative peeling converging in one iteration. Therefore, iterative peeling using an OPS will be as efficient as recursive peeling\(^1\) described in section 2.1.4. Note that an OPS may be used repeatedly e.g. for likelihood maximization.

### Approximation of the Likelihood in Pedigrees with Loops

In contrast to recursive peeling, iterative peeling is not restricted to non-looped pedigrees. In a complex pedigree with marriage or inbreeding loops, iterative peeling will implicitly cut loops and extend the pedigree with artificial members. To illustrate, consider the pedigree in Fig. 2.6 with a marriage loop. Using the connector sequence $[A \rightarrow B \rightarrow C \rightarrow D]$, the following computations will be executed:

\(^1\)Implicitly, recursive peeling is always following an OPS which depends on the starting individual.
2.1 Deterministic Maximum Likelihood

Figure 2.6: Pedigree consisting of four families with a marriage loop

1. first iteration:

(a) \([a_A(u_A) \text{ and } p_{A,B}(u_A)]\): using initialization values for \(a_B(u_B), p_{B,3}(u_B) \text{ and } p_{D,C}(u_D)\).

Figure 2.7: Virtual pedigree following step 1a)

(b) \([a_B(u_B), p_{B,A}(u_B), p_{B,3}(u_B)]\): using the initialization values for \(a_3(u_3), a_A(u_A), p_{C,D}(u_C) \text{ and } p_{D,C}(u_D)\), thus ignoring the mating \(C - D\).

Figure 2.8: Virtual pedigree following step 1b)
(c) \([a_C(u_C), p_{C,D}(u_C)]\): \(a_C(u_C)\) requires \(a_3(u_3), a_B(u_B)\) and \(p_{B, A}(u_B)\), the latter calculated in step 1b) ignoring the mating \([D - C]\). \(p_{C,D}(u_C)\) requires \(a_D(u_D)\) which is still in its initialization stage and thus ignores any ancestors to \(D\). This step cuts implicitly the pedigree and introduces the artificial founder \(D'\) as mate of \(C\). This detaches \(D\) from \(C\) as shown in Fig. 2.9.

![Figure 2.9: Virtual pedigree cut and extended following step 1c)](image)

(d) \([a_D(u_D), p_{D,C}(u_D)]\): \(a_D(u_D)\) requires \(a_A(u_A), a_B(u_B)\) and \(p_{B,3}(u_B)\). The latter was calculated in step 1b) ignoring posterior information to \(C\) (mating \([C - D]\)). \(p_{D,C}(u_D)\) requires \(a_C(u_C)\) which was calculated in step 1c) ignoring the mating \([C - D]\).

![Figure 2.10: Virtual pedigree cut and extended following step 1d)](image)
After the first iteration, the pedigree has been cut in step 1c) at member \( D \) and extended by introducing an artificial member \( D' \). Further pedigree extension is done in step 1d). Here, the anterior value \( a_D(u_D) \) and posterior value \( p_{D,C}(u_D) \) are calculated. These values are valid for \( D \) and \( D' \) as \( D' \) is nothing else than a second representation of \( D \). This attaches now anterior information to \( D' \) and posterior information through \( C'' \) to \( D \) as shown in Fig. 2.10.

This cut-extended pedigree could also be achieved by standard loop-cutting and extension of the pedigree without iterative peeling, the important difference is that no visual identification of the loop was necessary. Implicitly, the artificial members \( i', i'', \ldots \), have been assigned the same phenotypes as the original members \( i \). Further iterations will extend the pedigree step-wise as described in Wang et al. (1996) until the extensions will look like spirals (Fig. 2.11), the more iterations, the more turns in the pedigree.

**Figure 2.11:** Cut-extended pedigree after several iterations, anterior and posterior extensions drawn as coils

**Conditioning on Extensions**  This process of extending the pedigree can be continued infinitely. As described by Wang et al. (1996), the cut-extended pedigree can easily be split in the original part \( y_o \) and the extensions anterior (\( y_a \)) and posterior (\( y_p \)) to the connector where the cut has been made. The approximation for the likelihood can be improved
by *conditioning* on the extended part, i.e. computing the likelihood for the cut-extended pedigree and dividing by the likelihood of the extended parts:

\[
L(y_{\text{original}} \mid y_{\text{extended}}) = \frac{L(y_{\text{original}}; y_{\text{extended}})}{L(y_{\text{extended}})}
\]  

(2.21)

Wang et al. (1996) found empirically that it is not necessary to condition on both extensions available, i.e. it is sufficient to use one of the extension-coils shown in Fig. 2.11 for conditioning. In general, the likelihood for a pedigree with multiple and/or nested loops can now be approximated after loop-cutting and pedigree extension by:

1. Computing the likelihood for the entire cut-extended pedigree using terminal peeling.

2. Computing the likelihood for the extended parts i.e. by applying terminal peeling to the extension parts after setting the phenotypes of original individuals in the cut-extended pedigree to be missing.

3. Approximating the likelihood for a pedigree with loops by computing the likelihood for the cut-extended pedigree and dividing it by the likelihood for the extended parts.

**Optimum Loop-Cutting** Further improvement of the likelihood approximation can be achieved by optimization of the loop cutting. Within a loop, a pedigree can be cut and extended at any connector, but the amount of information yielded through the extended part of the pedigree may vary. Therefore, accuracy of approximation to the likelihood obtained from the cut-extended pedigree, depends for each loop on where the pedigree is cut and extended. Wang et al. (1996) proposed empirically determined rules to optimize loop cutting. The principle of this *optimum loop cutting* is to minimize the amount of artificially introduced information by the pedigree extensions.

As seen above, a family in a loop contains more than one connector. These connectors may be of two different types: connectors having an anterior *and* posterior families and
connectors having only posterior families. Wang et al. (1996) referred them to as type I and type II connectors. In the pedigree in Fig. 2.6, the connectors A, C, and D are of type I whereas connector B is of type II. A loop with a type I connector, say individual i', can be cut by replacing the connector in its posterior family or families with an artificial individual i', while leaving the original connector in its anterior family as a terminal member. Further pedigree extension as shown in section 2.1.5 will introduce artificial anterior member to i' as shown in Fig. 2.10 for D'. These artificially-introduced individuals collectively are called the anterior extension, and cutting a loop as described above is referred to as cutting with anterior extension.

In contrast, a loop with a type II connector is cut by replacing the connector in one of its posterior families that is part of the loop with an artificial founder individual i'. Starting at i', further pedigree extension as shown in section 2.1.5 will introduce artificial members posterior to i'. These artificially-introduced individuals are called the posterior extension, and cutting a loop as described above is referred to as cutting with posterior extension.

To minimize the amount of artificially introduced information by the pedigree extensions, the effect of a putative anterior or posterior extension on the genotype determination of a candidate i relative to the genotype determination for i by its own phenotype is measured. This is done by calculating

\[
\text{maxant}_{i} = \frac{\max_{u_{a}} \left[ \Pr(u_{i} \mid y_{a_{i}}) \right]}{\max_{u_{i}} \left[ \Pr(u_{i} \mid y_{i}) \right]}
\]

(2.22)

for each type I connector i and

\[
\text{maxpost}_{j} = \frac{\max_{u_{p_{j}}} \left[ \Pr(u_{i} \mid y_{p_{j_{i}}}) \right]}{\max_{u_{i}} \left[ \Pr(u_{i} \mid y_{i}) \right]}
\]

(2.23)

for each type 2 connector i through each mate j. Pr(u_{i} \mid y_{a}), Pr(u_{i} \mid y_{i}) and Pr(u_{i} \mid y_{p_{j}}) are computed as shown below from population genotype frequencies Pr(u) and from the
anterior, posterior and penetrance computed by iterative peeling:

\[
Pr(u_i | y_{a_i}) = \frac{a_i(u_i)}{\sum_u a_i(u_i)} \tag{2.24}
\]

\[
Pr(u_i | y_{i}) = \frac{Pr(y_i | u_i) Pr(u_i)}{\sum_u Pr(y_i | u_i) Pr(u_i)} \tag{2.25}
\]

and

\[
Pr(u_i | y_{p_j}) = \frac{\prod_{j} p_{ij} Pr(u_i)}{\sum_u \left[ \prod_{j} p_{ij} Pr(u_i) \right]} \tag{2.26}
\]

For individual \(i\), \(\max_{u_i}[Pr(u_i | y_{a_i})]\) measures the effect of the anterior extension on the genotype determination, \(\max_{u_i}[Pr(u_i | y_{p_i})]\) measures the effect of the posterior extension on the genotype determination, and \(\max_{u_i}[Pr(u_i | y_i)]\) measures the effect of the phenotype on the genotype determination. Therefore, \(\max_{post_j}\) measures the effects of the posterior extensions and \(\max_{ant}\) the effects of the anterior extensions relative to the genotype determination for individual \(i\) by its own phenotype.

The connectors for optimum loop cutting are determined for each loop by

1. searching for the type I connector with the smallest \(\max_{ant}\) value. If any full sibs of this connector has been used to cut a loop, the loop is cut at this connector without anterior extension; otherwise the loop is cut at this connector with anterior extension.

2. If no type I connector is found, searching for the type II connectors with the smallest \(\max_{post_j}\) value which has not yet been used for cutting another loop and cutting the loop at this connector with posterior extension.
2.1 Deterministic Maximum Likelihood

3. If type II connectors cannot be ranked, the loop is cut at an arbitrary type II connector with posterior extension.

2.1.6 Algorithm to Approximate the Likelihood for Complex Pedigrees

Wang et al. (1996) combined terminal peeling with loop cutting and pedigree extension by iterative peeling and conditioning on extensions and optimum loop cutting to an algorithm to approximate the likelihood for complex pedigrees.

In summary, this algorithm uses in a first step terminal peeling to peel away step by step all terminal families. This reduces the pedigree to the looped part. In the next step, iterative peeling is applied to the remaining looped part to extend the pedigree. Next, all loops are cut applying the optimum loop-cutting strategy. The likelihood of the now un-looped and extended pedigree is calculated subsequently by terminal peeling. Finally, the contributions of the artificial extensions is taken into account by conditioning the likelihood of the cut-extended pedigree on the likelihood of the extension parts, i.e. dividing the likelihood for the cut-extended pedigree by the likelihood for the extended parts.

**Detailed Algorithm** More in detail, the algorithm proposed by Wang et al. (1996) follows eight steps:

1. For each member \(i\) of the investigated pedigree:
   
   (a) initialization of the anterior and posterior value
   
   (b) calculation of the penetrance value and the corresponding log scaling factor.

2. Peeling off all terminal families by terminal peeling using an optimum peeling sequence (OPS). If the pedigree contains no loops, terminal peeling will reveal the
exact likelihood of the pedigree and no further calculations are necessary. If the pedigree contains loops, this step will reveal the looped part of the pedigree. The phenotypic information from the peeled terminal families is accumulated by the terminal peeling into either anterior or posterior values for their original connectors within the families in the looped part of the pedigree.

3. Extension of the looped pedigree part by iterative peeling as shown in Fig. 2.10 and 2.11.

4. Loop identification and cutting of these the optimum loop-cutting strategy.
   Peeling off the resulting terminal families using an OPS.
   If the pedigree cannot be reduced to a terminal family, i.e. some other loops are still in the pedigree, this step is repeated until all loops are cut.

5. Attach the anterior and posterior extensions from step 3 at the appropriate places to take into account the contributions of the extension parts to the likelihood:
   For each member $i$ in the cut-extended pedigree:
   
   (a) if $i$ was cut with anterior extension, then the anterior values of $i'$ is set to $a_i(u_i)$ from 3, the iteratively computed anterior value of $i$.

   (b) if $i$ was cut with posterior extension through mate $j$, then the posterior value of $i'$ is set to $p_{i,j}(u_i)$ from 3, the iteratively computed posterior values of $i$.

   (c) if $i$ was cut multiple times, e.g. one cut with anterior extension and one with posterior extension through mate $j$, then the anterior and posterior values are set to $a_i(u_i)$ and $p_{i,j}(u_i)$ respectively.

   Now a cut-extended pedigree has been generated.

6. The likelihood of the cut-extended pedigree is now calculated using again terminal peeling.
2.1 Deterministic Maximum Likelihood

7. To condition the likelihood of the cut-extended pedigree on the likelihood of the extension parts, the likelihood of the extensions is calculated by setting the phenotypes of all original individuals to be missing and repeating step 6.

8. Finally, the approximation for the likelihood is computed by $L(y_{original} \mid y_{extended})$ as shown in formula 2.21.

2.1.7 Summarizing

In the above, it has been shown that efficient deterministic Maximum Likelihood methods exist to approximate the likelihood of complex pedigrees, issue 1 of QTL mapping mentioned in section 1.3.

Unfortunately, this is not true for the issues 2 (number of loci in the model) and 3 (missing data).

As summations in deterministic methods are over unobserved phase known genotypes, i.e. consider all possible haplotype combinations a parent may transmit to its offspring, large numbers of loci in a model become computationally infeasible (issue 2) although the computational workload of peeling based approaches increases in general linearly to the number of individuals in the pedigree. This is even more the case if the genotype of some or many individuals is not observed (missing data, issue 3) as the computational workload is increasing exponentially to the amount of unobserved loci.

Additionally, in a mixed model of inheritance (Elston and Stewart, 1971; Morton and MacLean, 1974) where a major QTL segregates with several minor QTL, the conditional density of the phenotypic value given $u$, the vector of the joint genotypes at the QTL and the marker locus, cannot be written as in formula 2.1 because of the remaining segregating minor QTL. As there is currently no algorithm available to calculate exact likelihoods (Morton and MacLean, 1974; Elston and Stewart, 1971), approximations have to be used (Hasstedt, 1982; Hasstedt, 1991). Alternatively Bonney (1984; 1992) introduced a model based on familial correlations. Fernando et al. (1994) and Stricker et al. (1995b) introduced
the concept to model mixed inheritance with a mixture of discrete distributions originating from the segregation of a finite number of genetic loci.
To overcome some limitations of deterministic Maximum Likelihood, *Markov Chain Monte Carlo* (MCMC) methods have been proposed. Appropriate samplers can simulate realizations from models with any number of loci and any distribution assumed.

In a Markov chain, variables are repeatedly drawn from a state space $\theta$ of approximate distributions. Those draws are then corrected to approximate better their target conditional posterior distribution $p(\theta|y)$, given the current state of all other variables (Gelman et al., 1995). Sampling continues until the distributions of the variables converge to their true distribution. To assure convergence of the Markov chain, it is necessary (but not sufficient) to guarantee that samples are drawn from the entire state space $\theta$ of a variable. This means that the probability to reach from any state $i$ within $\theta$ any other state $j$ has to be non-zero ($Pr_{ij} > 0$). This property is called irreducibility.

The most common MCMC approach is the Metropolis-Hastings (Hastings, 1970) algorithm along with its two commonly used special cases; the Metropolis algorithm (Metropolis et al., 1953) and the Gibbs sampler (Geman and Geman, 1984). Given the target distribution $Pr(\theta|y)$, the original Metropolis algorithm creates a sequence of random points $(\theta^1, \theta^2, \ldots)$ by sampling at time $t$ candidate points $\theta^*$ from an approximate proposal distribution $J_t(\theta^*|\theta^{t-1})$ which has to be symmetric ($J_t(\theta_a|\theta_b) = J_t(\theta_b|\theta_a)$). The candidates are accepted with probability:

$$
\alpha = \min \left\{ \frac{Pr(\theta^*|y)}{Pr(\theta^{t-1}|y)}, 1.0 \right\}
$$

(2.27)

The Metropolis-Hastings algorithm generalizes the Metropolis algorithm by allowing and correcting for asymmetric proposal distributions leading to the acceptance probability:

$$
\alpha = \min \left\{ \frac{Pr(\theta^*|y)/J_t(\theta^*|\theta^{t-1})}{Pr(\theta^{t-1}|y)/J_t(\theta^{t-1}|\theta^*)}, 1.0 \right\}
$$

(2.28)
In the Gibbs sampler, the candidates are sampled from the full conditional distribution, hence ratio $\alpha$ is 1, leading to the acceptance of all candidates $\theta^*$. For efficient convergence to the target distribution, the samples from the proposal distribution need to move a reasonable distance in the state space and should not be rejected too frequently. Otherwise, the random walk moves slowly (slow mixing of the chain).

### 2.2.1 Sampling Genotypes

Sampling genotypes in a pedigree can be regarded as a Markovian process (Thomas and Cortessis, 1992) because a neighborhood system can be defined such that the genotype of an individual, conditional to its neighbors (relatives), is independent of all other pedigree members. This property makes MCMC methods such as the Gibbs sampler easy to implement.

MCMC approaches and especially Gibbs sampling were thought to be an important breakthrough in QTL mapping. However, it was soon realized that it is not evident to guarantee irreducibility of the Markov chain. This is illustrated in the example in Fig. 2.12 where genotypes are sampled at markers with more than two alleles.

![Diagram](image)

*Figure 2.12: Example of a reducible Markov chain generated by a single site Gibbs sampler: Two parents with unknown genotype and two offspring with genotype AB and CC, respectively.*

Single site Gibbs sampling will be locked into one of two possible initial configurations. The probability to move from one of both configurations shown in Fig. 2.13 to the other is zero. These two subspaces are called non-communicating subspaces and reducibility is due
to *vertical dependence*, i.e. dependence between individuals or more specifically between different individual’s genotypes.

![Figure 2.13: The two non-communicating initial configurations for the example in Fig. 2.12.](image)

It has been stated that this reducibility will not occur if at least one of the parent’s genotype is known, a situation often encountered in animal breeding where genotype information for male individuals is available. Cannings and Sheehan (2002) showed by a counter-example that this is not true in general (Fig. 2.14).

![Figure 2.14: Example for a reducible Markov chain generated by a single site Gibbs sampler when only a single allele of a single parent is missing. The two states of either allele B or C for “?” are non-communicating (Cannings and Sheehan, 2002).](image)

### 2.2.2 Sampling Blocks of Genotypes

To overcome these reducibility problems it has been proposed to update entire *blocks* of variables (genotypes for several individuals) jointly (Janss et al., 1995a; Jensen et al., 1993).
These blocks are typically formed by subfamilies of the pedigree. The efficiency of blocking depends largely on pedigree structure and the way how the blocks are built. In simple pedigrees like the granddaughter design, assigning individuals to a block is straightforward. For example, it will usually be advantageous to block sire and offspring. However, there is no algorithm to construct blocks that guarantee an irreducible Markov chain in large and complex pedigrees. Hence, blocking genotypes reduces in general the problem of reducibility but it does not eliminate it (Jensen and Kong, 1996), unless the size of the block is equal to the size of the pedigree. The latter approach has been proposed by Heath (1998) and Fernández et al. (2001).

2.2.3 ESIP Sampler

Fernández et al. presented an algorithm to sample genotypes that blocks the entire pedigree. In a first step, they apply the Elston-Stewart algorithm to peel the entire pedigree. In a second step, phase known genotypes are sampled using reverse peeling (Heath, 1998). If a pedigree is complex and cannot be peeled exactly by the Elston-Stewart algorithm, the pedigree is implicitly cut and extended by iterative peeling similar to the procedure of Wang et al. (1996) described in section 2.1.5. Samples are then generated from the cut-extended pedigree by an independence sampler and accepted with the probability given by the Metropolis ratio. Since the proposal distribution (i.e. the modified pedigree) is very close to the true pedigree, the sampler is expected to mix fast. Fernández et al. call this the ESIP-sampler due to combining Elston-Stewart with iterative peeling. Fernández et al. (2002) show that the ESIP-sampler reveals an irreducible and aperiodic Markov chain.

2.2.4 Sampling Allelic Origin

With respect to the issues of QTL mapping mentioned in section 1.3, efficiency of MCMC methods depends strongly on the size of the state space. In models with large numbers
of markers with numerous alleles, the number of possible genotypes for unobserved indi-
viduals becomes large (c.f. issue 2 and 3 in section 1.3 on page 4). Even a small number
of unobserved genotypes will lead to a huge state space to sample from and thus reveal
sampling to be inefficient. Of course, the same problem arises in deterministic Maximum
Likelihood where summations have to be carried out over unobserved genotypes. To give
an example, there are roughly 577 billion ($15^{10}$) possible genotypes in a model with 10
markers, each with 5 alleles, even when the phase of multiple heterozygous individuals is
not considered.

If the state space of a Markov chain becomes large, mixing of the Markov chain, i.e. the
ability of the chain to move freely within the entire state space may be poor. This is
especially the case if a chain moves only in small steps in its close neighborhood. A slow
mixing Markov chain may not converge to the true posterior distribution although the
requirements for irreducibility are fulfilled. In practice, slow mixing may have the same
effect as a reducible Markov chain and has therefore also been called *practical reducibility.*

In animal pedigrees, mixing is largely influenced by dependencies between homologous loci
of different individuals, e.g. of parents and their progeny, commonly referred to as *vertical
dependence* (Janss et al., 1995a) or by tight linkage between different loci, referred to as
*horizontal dependence.*

Individuals with observed genotypic information within a pedigree reduce the set of pos-
sible alleles for their relatives. In the example in Fig. 2.15, trying out possible alleles for
the unknown genes labeled by “?” is all but wasted effort unless the alleles are $A$ and $B$
respectively. The parents define the genotype of this individual completely, thus reducing
the state space to a smaller size. In this example it is even possible to exactly trace back
the origin of the genes of the terminal offspring. It can be derived from which parent the
individual with missing genotype received allele $A$ and allele $B$, thus defining a haplotype.
In a multi-allelic situation, allelic origin and thus phase known genotypes allow to derive
linkage information.
Figure 2.15: Allelic origin: Parent’s and offspring’s observed (unordered) genotypes limit the state space of possible alleles.

Thompson (1994) proposed to use MCMC on the state space on allelic origins. She distinguishes between the genotype for an individual and the origin of the alleles at that genotype. Allelic origin is the path of an allele from a parent to its offspring. It is also referred to as segregation indicator (Fig. 2.16).

Figure 2.16: Inheritance of alleles in pedigree of Fig. 2.15 represented by arrows. Each individual’s locus with its two alleles is represented by a dot. The allele on the left is assumed to be of maternal and on the right of paternal origin while the arrows represent the segregation indicators.

Segregation indicators define for non-founders whether a gene was inherited from the maternal grandmother, the maternal grandfather, the paternal grandmother or the paternal grandfather. However, they do not define which allele has been transmitted. Thus, gene flow can be separated in two parts, the paths that a founder gene takes as it is inherited
to offspring and, in addition to that, its allelic form. The paths of inheritance contain the complete information for making inference on the linkage between loci, the allelic form defines genotypic order. The state space for segregation indicators is usually considerably smaller (at most of equal size) than the state space of ordered genotypes, but provides still complete linkage information (Fig. 2.17, 2.18).

Figure 2.17: Example of a pedigree with identical heterozygous individuals only. The pedigrees represent two (of several possible) genotypic orders with equal likelihood.

Figure 2.18: Example in Fig. 2.17 represented by segregation indicators. The two orders in Fig. 2.17 map to a single state or descent graph.

Hence, sampling segregation indicators as proposed by Thompson reveals a potentially smaller state space than sampling of genotypes. In the algorithm proposed by Thompson and its extensions by Sobel and Lange (1996), segregation indicators for non-founders are sampled without sampling the genotypes for the founders. To do so, they first computed the joint probability of the founder genotypes and the non-founder segregation indicators. Then, the marginal probability for the segregation indicators is obtained by deterministically taking the sum of these probabilities over all founder genotypes that are compatible
with the observed data. The segregation indicators are sampled from this marginal distribution. In recent applications to livestock and plant breeding, the founder genotypes were also sampled (Jansen et al., 1998; Bink and Van Arendonk, 1999; Yi and Xu, 2001) using a single site Gibbs sampler. However, as mentioned above, single site Gibbs sampling does not guarantee an irreducible Markov chain in multi-allelic situations.

Sobel and Lange (1996) call the set of paths that the founder genes of a locus take as they descend through the pedigree a descent graph. Assuming a pedigree with $p$ members of whom $f$ are founders and $l$ observed marker loci, the descent graphs for all loci consists of $2lp$ nodes. For non-founder individuals, each node represents an allele of the respective locus of maternal or paternal origin, i.e. there is a maternal and a paternal node $n_i$ for each locus $l$. The segregation indicators are the arrows descending from a parental node $n_i$ to the homologue node in an offspring. They indicate as shown in Fig. 2.16 whether a parent contributes at a locus a grand-maternal or a grand-paternal allele to his offspring. Following segregation indicators, the origin of each node can be traced back to a node of a founder individual. All segregation indicators rooted at the same founder node form a descent tree incorporating exactly those nodes which inherit the corresponding founder allele (Sobel and Lange, 1996). There are $2lf$ such trees, but some may have length 0, i.e. some founder alleles are not inherited at all. A descent tree ties together all the offsprings of a founder which inherit the same allele, but there is no specification which allele is inherited. So, the arrows of the two descent trees in Fig. 2.18 may transmit either allele $A$ and $B$ or $B$ and $A$ respectively. If there is no genotyped individual within a descent tree, it cannot be determined which allele has been transmitted. If genotypic information about individuals within a descent tree exists, then the descent tree can only transmit alleles compatible with this information. If a descent tree for example passes through an individual which has the unordered genotype $\{A, B\}$ for the respective locus, the descent tree transmits either allele $A$ or $B$ as in Fig. 2.18. If the same descent tree also passes through some other individual with i.e. genotype $\{B, C\}$, it can transmit allele $B$ only and is therefore completely determined. A descent tree which passes through a homozygous individual becomes determined in the
same way. Thus, a descent tree may have either two or one allele permissible if it passes through one or more individuals with known genotype. If it passes through no individual with known genotype, all segregating alleles are permissible.

Non-consanguineous offspring inherit the two alleles at a locus from different founders. Hence, different descent trees meet in such offspring. If the offspring’s genotype is known, the corresponding descent trees become connected by the mechanism described above. If the two descent trees meet at an individual with the unordered genotype \( \{A, B\} \), both trees may still transmit either allele \( A \) or \( B \), but if one tree transmits \( A \), the other has to transmit \( B \) and vice versa. The two founder alleles of these trees are no longer independent and in a large pedigree, entire groups of founder alleles become dependent on each other (Fig. 2.19). Sobel and Lange (1996) call these groups connected components.

![Descent trees and connected components](image)

Figure 2.19: Descent trees and connected components: Each descent tree is labeled according to the founder node it is rooted in. Known genotypes are shown in braces. There are three connected components: (i) The descent trees rooted in founder nodes \( D \) and \( G \) with permissible alleles 1 and 2, respectively. (ii) The descent trees rooted in \( A \), \( B \) and \( C \). The only permissible allele for descent tree \( A \) is 1, for descent tree \( B \) is 3 and for descent tree \( C \) is allele 2. (iii) The descent tree rooted in \( F \) connected by a homozygous consanguineous individual. The permissible alleles are restricted to allele 1. The descent trees rooted in \( E \) and \( H \) do not pass through any genotyped individual. Thus all segregating alleles are permissible.

For a connected component the allelic information can be subsumed in an allele vector, i.e. a vector with one element for each founder node indicating the allelic type of the founder node. The genotyped individuals of the connected component induce dependencies between
the elements (founder nodes) of the allele vector. In Fig. 2.19, the connected component 
(i) consisting of the descent trees rooted in the founder nodes D and G are connected by
a single heterozygous \{1, 2\} individual, thus determining these founder nodes to allele 1
and 2 or vice versa. Therefore, there exist two allele vectors: the first containing allele 1
at the position corresponding to founder node D and allele 2 at the position corresponding
to founder node G, i.e. [1 2]. The second possible allele vector shows allele 2 at position
D and 1 at position G, i.e. [2 1]. Homozygous individuals in a connected component
determine completely the alleles at the corresponding founder node positions in the allele
vector. Thus, as soon as a homozygous individual is included in the connected component,
there is only a single allele vector. In general, a connected component may have no, one,
two or \(n\) allele vectors where \(n\) is the number of segregating alleles. If no allele vector or
\(n\) allele vectors exist, the connected component is either not compatible with the observed
data or not determined by observed data, i.e. does not include genotyped individuals.

### 2.2.5 Likelihood of Segregation Indicators

The joint likelihood of a descent graph \(\hat{G}\) and a marker phenotype vector \(M\) can be written
as the sum of all ordered genotype states \(G\) compatible with the segregation indicators of
\(\hat{G}\) and \(M\)

\[
\Pr(\hat{G} \cap M) = \sum_{G \in \hat{G} \cap M} \Pr(G) \tag{2.29}
\]

where \(G \mapsto \hat{G} \cap M\) denotes consistency between \(G\) and both \(\hat{G}\) and \(M\). Assuming Hardy-
Weinberg and linkage equilibrium, the probability of an ordered state \(G\) can be expressed as
the product of the probability of the alleles in \(G\) and the probability of their transmission.
The first is equivalent to the product over founder allele frequencies of the respective alleles
and can be designated as the prior probability \(\text{Prior}(G)\). In the single locus case, the latter
consists of the relevant powers of 0.5 for each time the founder allele is transmitted and
is referred to as transmission probability $\text{Trans}(G)$. In the multi-locus case assuming
no interference (Haldane mapping function), $\text{Trans}(G)$ is the product of the respective
recombination fractions $r$ and their complements $1 - r$ intervals separating the loci (Lange
and Sobel, 1991; Ott, 1991). If interference is assumed, $\text{Trans}(G)$ is the product of the
recombination values for the transmission pattern of each allele transmission (Ott, 1991).
All $G$ within a descent graph $\hat{G}$ have the same segregation indicators and thus equal
transmission probability. Hence, equation 2.29 can be written as

$$\Pr(\hat{G} \cap M) = \text{Trans}(\hat{G}) \sum_{G \to \hat{G} \cap M} \text{Prior}(G) \quad (2.30)$$

Under gametic equilibrium, each founder allele is sampled independently. Therefore,
$\sum_{G \to \hat{G} \cap M} \text{Prior}(G)$ is the sum of the products of founder allele probabilities of all per¬
missible founder allele combinations within $\hat{G}$. The connected components $C_1, \ldots, C_m$
of $\hat{G}$ with their corresponding allele vectors $a_1, \ldots, a_m$ contain exactly these permissible
founder allele combinations for $\hat{G}$. These combinations are drawn from the product of the
allele sets $S_1, \ldots, S_n$ for allele vector $a_i$ of connected component $C_i$ as the founder alleles
of different connected components are by construction independent.
For an ordered state $G$, $\text{Prior}(G)$ can thus be written as

$$\text{Prior}(G) = \prod_{i=1}^{m} \Pr(a_i) \quad (2.31)$$

where $a_i$ is the allele vector for connected component $i$ with $j$ permissible alleles $a_{ij}$ and
$\Pr(a_i)$ is $\prod_j \Pr(a_{ij})$. Rearranging equation 2.31 applying the distributive rule leads to
As mentioned above, there are either all or just two, one or no allele vector in a set $S_i$ of a connected component $i$. In the first case, no typed individuals are available and therefore $\sum_{a_i \in S_i} \Pr(a_i) = 1$. In the last case, no legal allele vectors exists and thus $\sum_{a_i \in S_i} \Pr(a_i) = 0$. In the remaining cases, the product formula $\Pr(a_i) = \prod_j \Pr(a_{ij})$ can be applied and the calculation of $\sum_{G \rightarrow G \cap M} Prior(G)$ can be performed easily. Combining equation 2.30 and 2.32 yields the following formula

$$
\Pr(G \cap M) = Trans(G) \prod_{i=1}^{m} \sum_{a_i \in S_i} \Pr(a_i).
$$

(2.33)

The likelihood of the pedigree in Fig. 2.19 can now be calculated following the above rearrangements:

1. determine the connected components $C_1, \ldots, C_m$ of this descent graph and the respective allele vectors $a_1, \ldots, a_m$, i.e.

(a) $C_1$, the descent trees rooted in founder alleles $D$ and $G$ has permissible alleles 1 and 2, respectively. $C_1$ thus has two allele vectors $a_{11} = [1 \ 2]$ and $a_{12} = [2 \ 1]$. 
(b) $C_2$, the descent trees rooted in $A$, $B$ and $C$. The only permissible allele for
descent tree $A$ is 1, for descent tree $B$ is 3 and for descent tree $C$ is allele 2, 
thus allele vector $a_2$ is $[1 \ 3 \ 2]$. 

(c) $C_3$, the descent tree rooted in $F$ connected by a homozygous consanguinous
individual leading to allele vector $a_3 = [1]$.

2. calculate $Trans(G)$ by counting all allele transmissions.

$$Trans(G) = \left(\frac{1}{2}\right)^{14}$$

3. calculate equation 2.32

$$\sum_{G \rightarrow M} Prior(G) = \prod_{i=1}^{3} \sum_{a_i \in S_i} \Pr(a_i)$$

$$= (a_1 a_2 + a_2 a_1)(a_1 a_3 a_2)(a_1)$$

where $a_i$ are the respective allele frequencies.

The descent trees rooted in $E$ and $H$ do not pass through any genotyped individual.
Thus all segregating alleles are permissible. The respective $\sum_{a_i \in S_i} \Pr(a_i)$ are 1 and
can thus be omitted.

4. Now combine the transmission and the prior part into

$$\Pr(\hat{G} \cap M) = \left(\frac{1}{2}\right)^{14}(a_1 a_2 + a_2 a_1)(a_1 a_3 a_2)(a_1)$$

2.2.6 Sampling Blocks of Segregation Indicators

Considering the reducibility problems of genotype sampling, the reduction in state space
offered by sampling of allelic origin was thought to be an important breakthrough in QTL
mapping. It was anticipated that this reduced state space may also solve the reducibility
Figure 2.20: Non-communicating states: Descent graph A

Figure 2.21: Non-communicating states: Descent graph C
problems associated with genotype sampling when more than 3 alleles are present. However, updating one single segregation indicator by switching the origin from the parental maternal node to the parental paternal node or vice versa (Thompson, 1994) may also lead to reducible Markov chains. In the presence of three and more alleles for a marker, different descent graphs may not communicate (Lange and Sobel, 1991) due to vertical dependence. This is shown in the example in Fig. 2.20 and 2.21: It is not possible to move between the two descent graphs compatible with the marker allele phenotypes in a finite number of steps applying a single-site Gibbs sampler for segregation indicators (Sobel and Lange, 1996).

As in genotype sampling described in section 2.2.1, the problem of reducible Markov chains may be overcome by updating blocks of variables jointly, i.e. updating blocks of segregation indicators. However, there is again no general rule how to build such blocks to eliminate reducible Markov chains unless the size of the block is equal to the size of the pedigree (Fernández et al., 2001).

Sobel and Lange (1996) proposed to execute a random number of transitions per updating step of the Markov chain instead of only one single transition (Sobel and Lange, 1993). This builds implicitly blocks of segregation indicators which are updated jointly and allows the Markov chain to step through one or several descent graphs which are not compatible with the observed marker phenotypes to get from one descent graph to another. This has been referred to as tunneling through incompatible states (Fig. 2.22). Using a random block size, which may be equal to the size of the pedigree, this procedure reveals an irreducible Markov chain because a single step can consist of an unlimited number of transitions and it is therefore possible to move from any descent graph to any other in a single updating step. However, using a random number of transitions per step of the chain independently of observed data, the proposed descent graph may be incompatible to the observed data and the sampler thus needs to correct for this. In other words, the candidate \( \theta^* \) is sampled at time \( t \) from the proposal distribution \( Q_t(\theta^*|\theta^{t-1}) \) which only approximates the true posterior distribution \( \text{Pr}(\theta|y) \). Hence, a Metropolis-Hastings algorithm has to be applied. Sobel and Lange (1996) show that the proposal matrix \( Q = q_{ij} \) is symmetric and positive.
Thus, moving from descent graph $a$ to graph $b$, $\Pr(\theta_a|\theta_b)$, is equally likely as moving from graph $b$ to $a$, $\Pr(\theta_b|\theta_a)$. Therefore, the Metropolis algorithm (Metropolis et al., 1953) can be applied and moving from a descent graph $\hat{G}_i$ to the descent graph $\hat{G}_j$ is accepted with the Metropolis probability

$$\alpha_{ij} = \min \left[ 1, \frac{\Pr(\hat{G}_j|M)}{\Pr(\hat{G}_i|M)} \right] = \min \left[ 1, \frac{\Pr(\hat{G}_j \cap M)}{\Pr(\hat{G}_i \cap M)} \right].$$

(2.34)

Thus, a more likely descent graph is always accepted, a less likely graph may be accepted or rejected and an incompatible graph is always rejected. The overall probability of moving from $\hat{G}_i$ to $\hat{G}_j$ is

$$p_{ij} = \begin{cases} q_{ij}\alpha_{ij} & \text{if } j \neq i, \\ 1 - \sum_{k \neq i} p_{ik} & \text{if } j = i, \end{cases}$$

(2.35)
where \( q_{ij} \) is the proposal probability \( \Pr(\hat{G}_j | \hat{G}_i) \). Because the proposals \( \theta^* \) are sampled from a positive and symmetric proposal distribution, i.e. \( q_{ij} \) is always positive, \( p_{ij} \) is also positive for all descent graphs \( \hat{G} \) compatible with \( M \). Therefore, the theoretical requirements for the Markov chain being aperiodic and irreducible are fulfilled (Sobel and Lange, 1996).

### 2.2.7 Building Blocks randomly

There are many ways to select the size and structure of blocks of segregation indicators, for example on the base of families or within connected components. Sobel and Lange (1996) proposed an approach where the blocks are re-built at random for each step of the Markov chain by the application of a sequence of transitions to several individuals. They show that the proposal distribution is symmetric if the choice of the sequence of transitions and the nodes where they are applied to is independent, while the selection of loci in multi-locus models may be correlated.

For the number of transitions, Sobel and Lange proposed a geometric distribution with mean 2, thus making a single transition per step has probability \( \frac{1}{2} \), two transitions have probability \( \frac{1}{4} \), three \( \frac{1}{8} \) an so on. Selection of the nodes where the transitions take place is random, but not necessarily uniform. Oversampling of certain nodes may provide a more efficient sampler. This may help the random walk of the Markov chain to take large steps within the state space and thus improve the mixing properties. However, the selection of nodes to be oversampled is not trivial in larger and complex pedigrees.

**Blocking within Families: Transition Rules**

Although building blocks of segregation indicators randomly ensures irreducibility of the Markov chain, it is usually inefficient because the proposals are frequently rejected due to the unconditional blocking and sampling. It may thus be often advantageous to group individuals according to relationship. Sobel and Lange (1996) proposed two grouping
strategies according to relationship within families.

**Figure 2.23:** Example of a source switch ($T_0$ transition rule). The white node is selected for the transition.

In these grouping strategies, sequences of simple source switches are applied to the offspring of an individual or of a couple. Sobel and Lange call a $T_0$-rule the switching of a segregation indicator from grand-paternal to grand-maternal origin or vice versa shown in Fig. 2.23.

**Figure 2.24:** Example of a $T_1$ transition. Both white nodes indicate the individual where the subtrees are detached, exchanged and re-attached.

**$T_1$ transition rule:** The first grouping strategy involves all offspring of an individual $i$ by detaching and exchanging the two descent subtrees of locus $l$ rooted in the two nodes of $i$ (Fig. 2.24). Hence, the relevant segregation indicators at locus $l$ of all offspring of $i$ are switched. More formally, if $i$ is a mother, a source switch $T_0$ is applied to the maternal node $l$ of all offspring of $i$. If $i$ is a father, a source switch $T_0$ is applied to the paternal node $l$ of all offspring of $i$. Thus, every offspring of $i$ which previously inherited $i$'s maternal
allele will now inherit i’s paternal allele and vice versa. Sobel and Lange call this block updating a $T_1$-rule.

**$T_2$ transition rule:** The second grouping strategy involves a couple $i$ and $j$ and their common offspring by detaching and exchanging the four descent subtrees of locus $l$ rooted in $i$ and $j$. Sobel and Lange call this block updating a $T_2$-rule. The $T_2$-rule has two variants:

![Figure 2.25: Example of a $T_{2a}$ transition. The white nodes indicate the couple selected for the transition.](image)

$T_{2a}$-rule: The subtrees rooted in $i$’s and $j$’s maternal node are detached and then exchanged, i.e. the subtree originally rooted at $i$’s maternal node is now rooted at $j$’s maternal node and vice versa. The two subtrees rooted in $i$’s and $j$’s paternal node are detached and exchanged analogously (Fig. 2.25).

![Figure 2.26: Example of a $T_{2b}$ transition. The white nodes indicate the couple selected for the transition.](image)

$T_{2b}$-rule: The subtrees rooted in $i$’s maternal and $j$’s paternal node are detached and then exchanged, i.e. the subtree originally rooted at $i$’s maternal node is now rooted at
j’s paternal node and vice versa. The two subtrees rooted in i’s paternal node and j’s maternal node are detached and exchanged analogously (Fig. 2.26).

The $T_2$ rule hence swaps paternally derived inheritance to maternally derived inheritance within a couple’s offspring. Fig. 2.25 and 2.26 indicate that $T_2$ transition rules can be expressed as a sequence of $T_1$ and $T_0$ rules as shown formally by Sobel and Lange (1996). Hence, every $T_1$ or $T_2$ transition can be seen as a sequence of source switches. These transition rules can therefore be applied like a single transition in sequence of transitions applied for random block building proposed by Sobel and Lange.

2.2.8 More Loci

In contrast to vertical dependence, i.e. dependence between loci in different individuals, horizontal dependence is due to dependence between different loci within the same individual. Descent graph sampling as described by Sobel and Lange blocks individuals randomly or randomly selects blocks of individuals, but it samples each locus individually. This strategy may be efficient for unlinked loci. However, segregation of linked loci shows dependencies which are not accounted for if loci are sampled individually. This may lead to poor mixing.

![Figure 2.27: Example of horizontal dependence in a three generation pedigree fully typed at two tightly linked loci. The segregation indicator arrows are omitted for simplicity.](image)
The example in Fig. 2.27 shows such a situation for two closely linked loci. All terminal offspring are genotyped \{1,3\} at the first and the second locus. The mother is double heterozygous \{1,2\}, the father doubly homozygous \{3,3\}. Both maternal grandparents are doubly heterozygous \{1,2\}. Thus, haplotype $\frac{3}{2}$ from the father and haplotype $\frac{1}{2}$ from the mother have been transmitted to the terminal offspring. The only unknown is the phase of the mother, i.e. whether she has genotype $\frac{1}{2}||\frac{3}{2}$, $\frac{2}{3}||\frac{1}{2}$, $\frac{1}{2}||\frac{1}{2}$ or $\frac{1}{2}||\frac{1}{2}$. Based on the loci being closely linked, the phase of the mother is likely either $\frac{1}{2}||\frac{1}{2}$ or $\frac{3}{2}||\frac{1}{2}$. Both these phases are equally likely due to non-informative grandparents. Hence, all offspring’s $\frac{1}{2}$-haplotypes are likely to be either from the maternal grandmother or from the maternal grandfather. For the descent graph sampler, it is required to execute a $T_1$-rule for both loci within the same proposal to switch between these two cases. For $n$ tightly linked loci, a sequence of $n$ $T_1$-rules would have to be applied to the same individual, this probability is virtually nil. The same is true for the unlikely case where the phase of the mother is $\frac{1}{2}||\frac{1}{2}$ or $\frac{1}{2}||\frac{1}{2}$.

### 2.2.9 Haplotype Sampling

To overcome slow mixing when segregation indicators at linked loci are sampled, a haplotype sampler accounting for linkage in the proposal is presented:

Instead of sampling segregation indicators for several loci individually, only the segregation indicators for an arbitrary locus $l$ are sampled using the transition rules by Sobel and Lange. Then, the haplotype for the remaining loci adjacent to the node which has been changed is sampled conditional to the segregation indicator of $l$ using a mapping function, i.e. based on the distance between the loci.

Thus, an individual’s haplotype is sampled combining transition rules at an arbitrary single locus with an appropriate mapping function accounting for interference. This can be illustrated by extending the above example in Fig. 2.27 by a third locus in close distance with identical genotypes as the other two loci. Assuming a $T_1$-transition at locus 2 in the mother, all offspring’s segregation indicators at this locus are changed from grand-maternal
to grand-paternal origin and vice versa. The segregation indicators for the adjacent loci 1 and 3 are now sampled according to probabilities derived from appropriate mapping functions taking into account distances between all loci.

In this example, mixing of the sampler between the likely phases \( \frac{1}{2} \parallel \frac{2}{2} \) and \( \frac{2}{2} \parallel \frac{1}{1} \) is much improved for tightly linked loci. Unfortunately, for sampling haplotypes, the proposal distribution \( J \) at time \( t \) for variable \( \theta \), \( J_t(\theta^*|\theta^{t-1}) \) is not symmetric, i.e. \( J_t(\theta_a|\theta_b) \neq J_t(\theta_b|\theta_a) \). As an example, assume two haplotypes \( \theta_a \) and \( \theta_b \) proposed by the above haplotype sampler for tightly linked loci. If haplotype \( \theta_a \) is a multiple recombinant while haplotype \( \theta_a \) is non-recombinant, the probability to move from \( \theta_a \) to \( \theta_b \) is larger than the probability to move from \( \theta_b \) to \( \theta_a \).

Therefore, the acceptance ratio \( \alpha \) can no longer be calculated using the Metropolis algorithm. To correct for the asymmetry in the proposal distribution, the acceptance ratio has to be calculated using the Metropolis-Hastings algorithm (Hastings, 1970)

\[
\alpha = \min \left\{ \frac{\Pr(\theta^*|y) / J_t(\theta^*|\theta^{t-1})}{\Pr(\theta^{t-1}|y) / J_t(\theta^{t-1}|\theta^*)}, 1.0 \right\}
\]  

(2.36)

where \( \Pr(\theta^*|y) \) is the likelihood for the proposed realization of the Markov chain of descent graphs and \( \Pr(\theta^{t-1}|y) \) is the likelihood for the previous realization of the Markov chain.

To calculate the probability to move from on descent graph to another, the moving probabilities probabilities for each individual \( j \) are considered sequentially, i.e.

\[
J_t(\theta^*|\theta^{t-1}) = \prod_{j=1}^{n} J_t(\theta^*_j|\theta^{t-1}_j)
\]

for moving from \( \theta^{t-1} \) to \( \theta^* \) and

\[
J_t(\theta^{t-1}|\theta^*) = \prod_{j=1}^{n} J_t(\theta^{t-1}_j|\theta^*_j)
\]

to move back. An individual’s haplotype is updated only if the origin of a maternal or
2.2 Markov Chain Monte Carlo Methods

paternal node $m_j$ has changed after one (or several) transition rules have been applied. If node $m_j$ has changed and thus a haplotype has been proposed, the deviation from symmetry needs to be calculated. Assume the change from the recombinant haplotype $0 1$ to the non-recombinant haplotype $0 0$ (where 0 stands for grand-maternal origin and 1 for grand-paternal origin of the allele) and let further $m_j = 2$, i.e. the transition rules by Sobel and Lange are applied at the second locus of individual $j$. This leads to the deviation from symmetry for $m_j$ caused by node $m_j = 2$ at individual $j$, $\Delta q_{m_j}$, to be calculated as

\[
\Delta q_{m_j} = \frac{J_t^I(\theta^*|\theta^{t-1})}{J_t^I(\theta^{t-1}|\theta^*)} = \frac{(1 - r_1)(1 - r_2)}{r_1 r_2}
\]

Summation of the deviation for both nodes $m$ in all individuals leads to detailed balance:

\[
\alpha = \min \left\{ \frac{\Pr(\theta^*|y) / \prod_{j} \Delta q_{m_j}}{\Pr(\theta^{t-1}|y)}, 1.0 \right\}.
\]  

(2.37)

If the origin of node $m_j$ has not changed and thus no haplotype has been sampled, symmetry is maintained and

\[
\Delta q_{m_j} = \frac{J_t^I(\theta^*|\theta^{t-1})}{J_t^I(\theta^{t-1}|\theta^*)}
\]

cancels out in equation 2.36. Therefore, the product $\prod_j$ in (2.37) needs only to be taken over the individuals $j$ with one of their nodes updated during the proposal realization.

2.2.10 Cascading Origin Sampler

The proposal distribution generated by the haplotype sampler may not always be a close approximation of the true posterior distribution. In the above examples with tightly linked loci, it is evident that the haplotype sampler will rarely propose a recombinant haplotype.
Additionally, a recombinant haplotype will have low acceptance ratio in the above examples. Expanding the example in Fig. 2.27 with two likely recombinant offspring as shown in Fig. 2.28 will reveal slow mixing for the haplotype sampler.

Based on data and the tight linkage of the loci, the offspring $\frac{1}{2}||\frac{3}{3}$ and $\frac{1}{2}||\frac{3}{3}$ are very likely to be recombinant. However, it is unlikely for the haplotype sampler to propose these recombinants. E.g. whenever a Ti-transition is executed at locus $l$ for the mother, all maternal haplotypes at that locus for all offspring are switched to the other grand-paternal origin. The rest of the haplotype is proposed based on a mapping function revealing a low probability for recombinants at tightly linked loci. Thus, the sampler will be locked into one descent graph configuration by the two additional offspring.

To overcome this horizontal dependence, the cascading origin (CO-) sampler is proposed. In the cascading origin sampler, a node $m$ at individual $j$ is chosen to apply some transition rule as in the haplotype sampler. Then, in contrary to the haplotype sampler, the same transition rule is applied to the adjacent loci depending on a mapping function, i.e. accounting for interference between loci. The word cascading refers to this process being executed sequentially over all nodes towards both ends of the haplotype starting from the initial locus.

In the above example (Fig. 2.28), the CO-sampler is likely to exchange the origin of all offspring’s loci at once whenever a $T_1$-transition is applied to an arbitrary locus $l$ of the

**Figure 2.28: Example of horizontal dependence not solved by the haplotype sampler.**
mother, i.e. all $1\|3$ offspring are likely to change into $2\|3$, all $2\|3$ likely into $1\|3$, offspring $2\|3$ likely changes to $1\|3$ and offspring $1\|3$ to $2\|3$. This leads to a freely mixing Markov chain.

The cascading origin sampler has symmetric proposal distributions, i.e. $J_i(\theta_a|\theta_b)$ is equal to $J_i(\theta_b|\theta_a)$. In contrast to the haplotype sampler (section 2.2.9 on page 56), the change from the recombinant haplotype $0\|1$ to the non-recombinant haplotype $0\|0$ (where 0 stands for grand-maternal origin and 1 for grand-paternal origin of the allele) leads to no deviation from symmetry. In tightly linked loci, the CO-sampler is likely to propose the change from $0\|1$ to $0\|1$; the change from $0\|1$ to haplotype $0\|0$ and vice versa requires the CO-sampler twice to not apply the same transition rule. These two cases are equally (un-)likely. Therefore, the Metropolis algorithm can be applied and there is no need to calculate $\Delta q_n$.

Hence, for linked loci, the CO-sampler preserves certain patterns of inheritance at haplotypes, i.e. the pattern of recombinant and non-recombinant inheritance at the loci within a haplotype is preserved at the expense of changing allelic origin. The haplotype sampler in contrary is likely to assign the same origin to all adjacent loci as was proposed for the locus initially updated by transition rules. In this respect, the haplotype sampler is preserving origin.
Chapter 3

Combining Deterministic Calculation and Stochastic Simulation

The major goal of this research project was to combine the stochastic approach of Thompson and Sobel and Lange with deterministic Maximum Likelihood (ML) methods (c.f. section 1.4 on page 5). This combined algorithm shall be integrated in the computer program package MATVEC (Wang et al., 2002). The basic idea was to use peeling at a biallelic QTL while the inheritance at multiple markers is evaluated by a Markov chain of segregation indicators. Since deterministic ML is retained for inference about the QTL, the problem of complex pedigree needs to be addressed. This is done by the approach of Wang et al. (1996) (c.f. section 2.1.6 on page 31).

3.1 MATVEC

The computer program package MATVEC was originally developed by Wang et al. (2002). It is a highly integrated program package for animal breeders written in the object-oriented programming language C++. It offers routines and algorithms for genetic evaluation and
variance component estimation under various linear and non-linear models (Kachmann and Fernando, 2002).

MATVEC incorporates a library of C++ classes and an environment for interactive statistical analysis. The program classes represent a powerful yet flexible set of tools adapted for animal breeding problems. Object-oriented programming represents a way of thinking and a methodology for computer programming that is quite different from the traditional approaches supported by structured programming languages. Large tasks are divided into several small subtasks which are treated by standardized modules (objects). These objects are designed to be flexible and unspecific regarding their input and standardized in their output. These features allow to treat a vast amount of tasks by assembling an appropriate set of objects from a library. Hence, object-oriented programming makes problem solving a more human-like activity and increases the re-usability of software code.

In an object-oriented programming environment, the flexibility and standardization of the objects is crucial. In the final program, different objects are combined into a program. They act like black boxes tackling specific subproblems. To get the pieces or objects working together, the interfaces have to be specified carefully to provide flexible and reusable objects. In the object-oriented programming approach, an object is treated as single variable although it may contain itself many different variables. This property allows to create very easily large numbers of objects with the same properties.

C++ provides object-oriented capability without loss of run-time or memory inefficiency and is available for almost every computer system from PC to mainframe. MATVEC is written in ASCII C++, the executable programs were compiled using gnu gec 3 compilers.

3.1.1 Objects in MATVEC

In MATVEC, the objects group informations and methods used in animal breeding in a hierarchical structure. To give an example, the objects Individual and Population are described below.
Object Individual  All informations and methods specific to an individual are grouped in the object Individual. There is one such object for each individual. The Individual object stores, among other information, the individual’s identity and sex, its parents and offspring, its phenotype and genotype, penetrance values, anterior and posterior values, segregation indicators. All methods necessary to access these variables are also part of the Individual object (encapsulation).

Once all desired variables for the Individual object are defined, a simple command allows to build vectors of Individual objects. Variables within an object can be accessed by pointers, i.e the storage address of the relevant variable. This allows for example to define the variables mymother and myfather in the Individual object i, which are pointers to the Individual objects of i’s mother j and father k. With these pointers it is possible to access all variables of j and k. For example, within the Individual object the operation \( \rightarrow \text{mymother} \rightarrow \text{mymother.id} \) returns i the identity of i’s maternal grandmother.

Object Population  The object Population refers to all informations relative to all individuals in a pedigree, e.g. the number \( n \) of individuals, the genetic model (number of chromosomes, properties of markers and QTL, variances, peeling sequences etc.). A Population object holds \( n \) Individual objects as described above. It further includes methods to read, access and process data.

The members of a Population object may be accessed by a pointer to its address. Taking the same example as above, within the Population object individual i is accessible as member(i). Hence individual i’s maternal grandmother identity is accessed within the Population object by \( \rightarrow \text{member}(i) \rightarrow \text{mymother} \rightarrow \text{mymother.id} \).
3.2 Segregation Indicator Sampler

The segregation indicator sampler for multiple markers has been incorporated in the object oriented philosophy of *MATVEC* using the objects described above. This allowed to adapt existing basic program tools, e.g. for reading and verification of pedigree- and data files. With a few exceptions, all methods and variables for the sampler have been integrated in the *Population* and *Individual* object classes.

The program flow of the segregation indicator sampler implemented in *MATVEC* can be subdivided in three major components:

1. Setup

2. Calculation of the likelihood of a set of segregation indicators

3. The segregation indicator sampler.

### 3.2.1 Setup

*MATVEC* provides the three object classes *GeneticDist*, *Pedigree* and *Data* to read in and validate variables. All variables concerning the genetic structure valid for all individuals are read in the *GeneticDist* object, i.e. the number of chromosomes, the number and position of markers, the number of alleles at each marker, the respective allele frequencies, the QTL position and the genotypic values for its different genotypes. Pedigree data, i.e. mother and father, and phenotypic data are read into the objects *Pedigree* and *Data*.

The *Population* object is built using the above input data. Subsequently, the *Individual* objects for each individual in the pedigree are created. Individual ID are recoded such that offspring do not precede their parents. The necessary memory space for each *Individual* object is deduced from the genetic structure. After these steps, the *Population* object holds an *Individual* object for each individual in the pedigree. Each *Individual* object
holds among other information its marker and QTL phenotype, the pointer addresses of
the individual's parents, etc.
The Individual object has a binary variable indicating whether it is a copy of the par¬
ent's maternal or paternal allele for the maternal and the paternal allele at each marker
locus. In other words, these two variables represent for each locus the inheritance of the
maternal and the paternal allele, i.e. a segregation indicator. A segregation indicator of
0 or 1 refers to grand-maternal or grand-paternal descendance of an allele, respectively.
In addition to the segregation indicator variables, each Individual object holds for each
marker two founder allele variables indicating the specific founder allele from which the
two alleles originate, i.e. are identical by descent (IBD) to. For founder individuals, these
variables are numbered sequentially during the initialization.

3.2.2 Initial Configuration

The initial configuration of the segregation indicators, i.e the initial descent graph, has
to be compatible with the observed marker data. The initial configuration of segregation
indicators needs to be provided to the program and is read into the Individual objects
through method input_descentGraph.

Although all legal descent graphs \( \tilde{G} \) communicate in theory through the approach shown
in section 2.2.4, in practice there may exist bottlenecks very unlikely to be passed through.
Hence it is important to start from an initial configuration as close as possible to the true
state of nature. Of course finding such an initial configuration is closely related to finding
the most likely marker haplotype vector for the underlying pedigree. Although an initial
configuration may be derived by hand for small pedigrees, this task may be very compli¬
cated for complex pedigree with many markers. Sobel and Lange (1996) proposed the use
of an iterative genotype-elimination algorithm to build an initial descent graph \( \tilde{G} \). An
other way to get a legal initial descent graphs \( \tilde{G} \) is the use of the ESIP sampler (Fernández
et al., 2001) described in section 2.2.3 (page 38) already implemented in MATVEC. This
latter approach has been adopted for the purpose of the examples provided here.
3.2.3 Calculation of the Likelihood of a Descent Graph

The (log-) likelihood of a descent graph is calculated in the Population object member function `descent_graph_log_lhood()`. The function follows the theory presented in section 2.2.5.

Each time the function is called, the founder allele variables of all offspring are set first. This is done by tracing recursively the segregation indicators back to the respective founder allele. After this initialization, the prior probability \( \sum_{G \in \hat{G}^M} \text{Prior}(G) \) is calculated by the function `calc_prior_descent_graph` for each marker. In this function, the connected components are built first, i.e. dependent founder trees are grouped by the function `build_connected_groups` (c.f. Fig. 2.19).

The allele vector(s) for the respective connected groups are subsequently built using the function `build_allele_vector`. The permissible alleles for the different allele vectors are set by the function `update_allele_vector`.

Once the allele vector is built, the prior probability \( \text{Prior}(G) = \prod_{i=1}^{m} \text{Pr}(a_i) \) for the respective marker locus is calculated. The overall prior probability is calculated by the summation of the log’s of all graph probabilities.

The transmission probability is calculated in a loop including all non-founder individuals. If only a single marker is calculated, each transmission of an allele has probability \( \frac{1}{2} \). In multi-allelic cases, the transmission probability is calculated depending on recombination between loci using the recombination fraction \( r \) or \( 1-r \) respectively. The map function used can be exchanged, default is the Haldane map function (Ott, 1991), another multi-locus feasible map function would be the binomial.

3.2.4 The Segregation Indicator Sampler

Sobel & Lange sampler For the Sobel & Lange segregation indicator sampler, one or several transitions upon the descent graphs are proposed and evaluated using the Metropo-
3.2 Segregation Indicator Sampler

is probability $\alpha$ as described in section 2.2.6 (page 50). This is done in the basic form of the function `M.H_sample_ext` of the `Population` object.

The first step within this function is to randomly select an individual and a marker locus where a transition is to be applied. The second step is to select one of the four transition rules described by Sobel and Lange. The selected rule is subsequently applied if it is applicable to the selected individual (e.g., application of another rule than $T_0$ to a terminal is not possible). If a combined transition rule ($T_1$ and $T_2$) is selected, the appropriate sequence of $T_0$ rules is executed (c.f section 2.2.7).

The number of times these two steps are executed to obtain the next realization of a descent graph, i.e., the number of transitions, is drawn from a geometric distribution with mean 2. Thus a single transition has probability $\frac{1}{2}$, two transitions have probability $\frac{1}{4}$, three $\frac{1}{8}$, and so on. Proposed descent graphs are always accepted if they are more likely to the old descent graphs. If they are less likely, they may be accepted or rejected and incompatible graphs are always rejected. If the proposal is rejected, all segregation indicators are set back to their previous state.

The Haplotype (HT-) Sampler  To overcome slow mixing if segregation indicators are sampled at tightly linked loci, a haplotype sampler accounting for linkage in the proposal distribution as described in section 2.2.9 has been implemented in the function `sample_self` of the `Individual` object.

Instead of sampling segregation indicators for several loci individually, only the segregation indicators for an arbitrary locus $l$ are sampled using the transition rules by Sobel and Lange. Then, the haplotype for the remaining loci adjacent to the node which has been changed is sampled conditional on the segregation indicator of $l$ using a mapping function (default is the Haldane map function (Ott, 1991)), i.e. based on the distance between the loci. Thus, an individual’s haplotype is sampled combining the transition at an arbitrary single locus with a appropriate mapping function accounting for interference.

To account for the non-symmetric proposal distribution (c.f. 2.2.9), the acceptance ratio
Combining Deterministic Calculation and Stochastic Simulation

\( \alpha \) is calculated using the Metropolis-Hastings algorithm. To calculate the probability to move from descent graph \( \theta^{t-1} \) to \( \theta^* \), the moving probabilities for each individual \( j \) are considered sequentially, i.e. \( P_t(\theta^*|\theta^{t-1}) = \prod_{j=1}^{n} P_t(\theta_j^*|\theta_j^{t-1}) \) for moving from \( \theta^{t-1} \) to \( \theta^* \) and \( P_t(\theta^{t-1}|\theta^*) = \prod_{j=1}^{n} P_t(\theta_j^{t-1}|\theta_j^*) \) to move back. An individual's haplotype is updated only if the origin of a maternal or paternal node \( m_j \) has changed after one (or several) transition rules have been applied.

The Cascading Origin (CO-) Sampler To overcome horizontal dependence, the cascading origin (CO-) sampler as described in section 2.2.10 (page 57) has been implemented in the function `apply_SL` of the `Individual` object. In the cascading origin sampler, a node \( m \) at individual \( j \) is chosen to apply some transition rule as in the haplotype sampler. Then, in contrary to the haplotype sampler, the same transition rule is applied to the adjacent loci depending on a mapping function, i.e. accounting for interference between loci (default is the Haldane map function (Ott, 1991)). The word cascading refers to this process being executed sequentially over all nodes towards both ends of the haplotype starting from the initial locus.

The cascading origin sampler has symmetric proposal distributions, i.e. \( P_t(\theta^{t-1}|\theta^*) \) is equal to \( P_t(\theta^*|\theta^{t-1}) \). In contrast to the haplotype sampler (section 2.2.9 on page 56), the change from the recombinant haplotype \( \begin{pmatrix} 0 \\ 1 \end{pmatrix} \) to the non-recombinant haplotype \( \begin{pmatrix} 0 \\ 0 \end{pmatrix} \) (where 0 stands for grand-maternal origin and 1 for grand-paternal origin of the allele) leads to no deviation from symmetry.
3.3 Applications

The MATVEC implementation of the segregation indicator sampler, i.e. the Sobel & Lange (SL-) sampler (Sobel and Lange, 1996), the haplotype (HT-) sampler and the cascading origin (CO-) sampler can easily be adapted to provide estimates of allele or genotype frequency and marker phase information to (iteratively) peel a QTL (combining MCMC- and deterministic maximum likelihood methods).

The different samplers (S&L, HT and CO) can be applied separately or combined. Combination of the different samplers, i.e. executing a predefined number of rounds $k_1$ of S&L-samples, $k_2$ rounds of HT-samples and $k_3$ of CO-samples, may be the key to fast and reliable mixing properties of the Markov chains (Stricker et al., 2002).

Recently, the above samplers have been applied to estimate probabilities of descent for QTL alleles (PDQ). PDQ’s are essential for the computation of identical by descent- (IBD-) probabilities at a QTL given marker information. In marker assisted best linear unbiased prediction (BLUP), IBD probabilities are needed to construct Henderson’s Mixed Model Equations (HMME).

3.4 Performance of Segregation Indicator Samplers

To assess the performance of the algorithms, marker inheritance in several hypothetic pedigree has been considered.

3.4.1 Single Nuclear Family

To show some properties of the segregation indicator samplers, consider a pedigree consisting of a nuclear family with 35 offspring. Assume the genotype for 34 of the offspring
3 Combining Deterministic Calculation and Stochastic Simulation

Figure 3.1: Nuclear family with 35 offspring. The genotype of the parents and one offspring is unknown. 17 offspring are heterozygous (Aa) and 17 are homozygous (AA).

Figure 3.2: Nuclear family of Fig. 3.1 in segregation indicator representation: a legal initial configuration.

is known: 17 are heterozygous \{Aa\} and 17 are homozygous \{AA\} (Fig. 3.1 and Fig. 3.2). The allele frequencies are 0.75 and 0.25 for allele A and a, respectively. The genotype for the parents and one offspring is unknown. The Sobel & Lange (SL-) segregation indicator sampler was used to sample the missing genotypes.

Table 3.1 gives the exact probabilities for the missing genotypes being \{aa\}, \{Aa\} or \{AA\} and the estimated probabilities after 10'000, 100'000 and one million samples drawn. The computing time for this simulation on a 350 MHz INTEL Pentium II processor running LINUX 2.4 were 13, 70 and 674 seconds, respectively.
3.4 Performance of Segregation Indicator Samplers

<table>
<thead>
<tr>
<th>samples</th>
<th>$\Pr{aa}$</th>
<th>$\Pr{Aa}$</th>
<th>$\Pr{AA}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>mother</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^4$</td>
<td>0.0</td>
<td>0.4724</td>
<td>0.5276</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.0</td>
<td>0.49248</td>
<td>0.50752</td>
</tr>
<tr>
<td>$10^6$</td>
<td>0.0</td>
<td>0.492223</td>
<td>0.507777</td>
</tr>
<tr>
<td>exact prob.</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>father</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^4$</td>
<td>0.0</td>
<td>0.5276</td>
<td>0.4724</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.0</td>
<td>0.50752</td>
<td>0.49248</td>
</tr>
<tr>
<td>$10^6$</td>
<td>0.0</td>
<td>0.507777</td>
<td>0.492223</td>
</tr>
<tr>
<td>exact prob.</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>offspring</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$10^4$</td>
<td>0.0</td>
<td>0.5443</td>
<td>0.4557</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.0</td>
<td>0.49799</td>
<td>0.50201</td>
</tr>
<tr>
<td>$10^6$</td>
<td>0.0</td>
<td>0.499928</td>
<td>0.500072</td>
</tr>
<tr>
<td>exact prob.</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 3.1: Estimated marginal probabilities obtained by the segregation indicator sampler and exact marginal probabilities for the individuals in with unknown genotype of the nuclear family in Fig. 3.1 and Fig. 3.2

3.4.2 Half-sib Family

Similar to the above example, the S&L-sampler was used to estimate probabilities in a large half-sib family (Fig. 3.3). The allele frequencies are 0.75 and 0.25 for allele A and a, respectively. The pedigree consists of 3 founders: one sire and two dams, where each family has again 35 offspring. In both nuclear families the genotype for for 34 offspring is known, 17 are homozygous ($AA$) and 17 heterozygous ($Aa$). The genotype of the parents and of one offspring in each nuclear family is unknown. Four different initial descent graph configurations were used to obtain estimates for this pedigree by the original segregation indicator sampler (Fig. 3.4 to Fig. 3.7). The genotype probabilities were also calculated exactly by SALP (Stricker et al., 1995c). The computing time for the simulation of one million legal samples was 1510 seconds on a 350 MHz INTEL Pentium II running LINUX 2.4.
Table 3.2: Estimated marginal probabilities obtained by the segregation indicator sampler and exact marginal probabilities obtained by SALP for the individuals in with unknown genotype of the half-sib family in Fig. 3.3 (estimates are from $10^6$ samples).
Figure 3.3: Large half-sib family with 70 offspring. In each family the genotype of the parents and of one offspring is unknown, 17 offspring are heterozygous (Aa) and 17 are homozygous (AA).

Figure 3.4: Half-sib family of Fig. 3.3 in segregation indicator representation. Initial configuration 1 with heterozygous females 1 and 3.

Figure 3.5: Half-sib family of Fig. 3.3 in segregation indicator representation. Initial configuration 2 with heterozygous male 2.
Figure 3.6: Half-sib family of Fig. 3.3 in segregation indicator representation. Initial configuration 2 with heterozygous individuals 2 and 3.

Figure 3.7: Half-sib family of Fig. 3.3 in segregation indicator representation. Initial configuration 2 with heterozygous individuals 1, 2 and 3.
3.4 Performance of Segregation Indicator Samplers

3.4.3 Pedigree with Loops

To assess the performance of the segregation indicator sampler in a pedigree with loops, genotype probabilities were estimated in a hypothetical ABO-blood-type pedigree used by Sheehan and Thomas (1993) depicted in Fig. 3.8. Applying the original segregation indicator sampler, three different chain lengths were used to obtain the estimates: 10’000, 100’000 and 1’000’000. The computing time for these chains were 17, 157 and 1517 seconds. The results were compared to the true marginal probabilities (Table 3.3 and Table 3.4). The mean differences for the three chain sizes were $3.6 \times 10^{-3}$, $2.2 \times 10^{-3}$ and $1.8 \times 10^{-3}$, respectively, with largest differences of $6.2 \times 10^{-2}$, $3.2 \times 10^{-2}$ and $3.2 \times 10^{-2}$ (Fernández et al., 2002). Table 3.3 and 3.4 show that convergence to the true marginal genotype frequencies for the individuals with unknown genotypes is rapidly achieved. The S&L-sampler is thus not sensitive to the pedigree structure except if strong vertical dependence is present.

Figure 3.8: Test pedigree with hypothetical ABO-blood-type (Sheehan and Thomas, 1993).
<table>
<thead>
<tr>
<th>Id</th>
<th>exact</th>
<th>0.06702</th>
<th>0.10052</th>
<th>0.30052</th>
<th>0.03351</th>
<th>0.26754</th>
<th>0.23089</th>
</tr>
</thead>
<tbody>
<tr>
<td>0^4</td>
<td>0.0638</td>
<td>0.0957</td>
<td>0.2657</td>
<td>0.0319</td>
<td>0.2595</td>
<td>0.2534</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.063163</td>
<td>0.094745</td>
<td>0.294745</td>
<td>0.0315815</td>
<td>0.27908</td>
<td>0.257859</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0631067</td>
<td>0.09466</td>
<td>0.29666</td>
<td>0.0315533</td>
<td>0.277767</td>
<td>0.258253</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>4</td>
<td>0.02241</td>
<td>0.02241</td>
<td>0.24482</td>
<td>0.00560</td>
<td>0.12241</td>
<td>0.58236</td>
</tr>
<tr>
<td>0^4</td>
<td>0.02236</td>
<td>0.02236</td>
<td>0.24652</td>
<td>0.005815</td>
<td>0.12326</td>
<td>0.578885</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.0230964</td>
<td>0.0230964</td>
<td>0.246193</td>
<td>0.0057741</td>
<td>0.123096</td>
<td>0.578744</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0234303</td>
<td>0.0234303</td>
<td>0.246861</td>
<td>0.00585758</td>
<td>0.12343</td>
<td>0.576991</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>5</td>
<td>0.01558</td>
<td>0.02336</td>
<td>0.22245</td>
<td>0.00779</td>
<td>0.18200</td>
<td>0.54882</td>
</tr>
<tr>
<td>0^4</td>
<td>0.01843</td>
<td>0.027645</td>
<td>0.231595</td>
<td>0.009215</td>
<td>0.192475</td>
<td>0.52084</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.019133</td>
<td>0.0286995</td>
<td>0.232748</td>
<td>0.0095665</td>
<td>0.194117</td>
<td>0.545737</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0195404</td>
<td>0.0295106</td>
<td>0.232499</td>
<td>0.0097702</td>
<td>0.193702</td>
<td>0.545428</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>7</td>
<td>0.02241</td>
<td>0.02241</td>
<td>0.24482</td>
<td>0.00560</td>
<td>0.12241</td>
<td>0.58236</td>
</tr>
<tr>
<td>0^4</td>
<td>0.024832</td>
<td>0.024832</td>
<td>0.249664</td>
<td>0.006208</td>
<td>0.124832</td>
<td>0.569632</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.0253624</td>
<td>0.0253624</td>
<td>0.247125</td>
<td>0.0058906</td>
<td>0.123562</td>
<td>0.576297</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0253058</td>
<td>0.0253058</td>
<td>0.246812</td>
<td>0.0058515</td>
<td>0.123406</td>
<td>0.577132</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>8</td>
<td>0.00995</td>
<td>0.00995</td>
<td>0.19424</td>
<td>0.00249</td>
<td>0.09712</td>
<td>0.68826</td>
</tr>
<tr>
<td>0^4</td>
<td>0.0111</td>
<td>0.0111</td>
<td>0.2074</td>
<td>0.00275</td>
<td>0.10373</td>
<td>0.66406</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.0105648</td>
<td>0.0105648</td>
<td>0.2016</td>
<td>0.0026412</td>
<td>0.1008</td>
<td>0.67383</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0106902</td>
<td>0.0106902</td>
<td>0.20211</td>
<td>0.0026726</td>
<td>0.101055</td>
<td>0.672782</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>11</td>
<td>0.0</td>
<td>0.0</td>
<td>0.16702</td>
<td>0.0</td>
<td>0.11099</td>
<td>0.72199</td>
</tr>
<tr>
<td>0^4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.1746</td>
<td>0.0</td>
<td>0.1164</td>
<td>0.7091</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.175735</td>
<td>0.0</td>
<td>0.117925</td>
<td>0.70634</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.176926</td>
<td>0.0</td>
<td>0.118801</td>
<td>0.704273</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>13</td>
<td>0.0</td>
<td>0.25</td>
<td>0.25</td>
<td>0.0</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>0^4</td>
<td>0.0</td>
<td>0.1794</td>
<td>0.2859</td>
<td>0.0</td>
<td>0.2859</td>
<td>0.2488</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.0</td>
<td>0.24935</td>
<td>0.25485</td>
<td>0.0</td>
<td>0.25485</td>
<td>0.24168</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0</td>
<td>0.247365</td>
<td>0.25324</td>
<td>0.0</td>
<td>0.25324</td>
<td>0.246154</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>14</td>
<td>0.0</td>
<td>0.0</td>
<td>0.16702</td>
<td>0.0</td>
<td>0.11099</td>
<td>0.72199</td>
</tr>
<tr>
<td>0^4</td>
<td>0.0</td>
<td>0.0</td>
<td>0.17504</td>
<td>0.0</td>
<td>0.11792</td>
<td>0.70704</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.0</td>
<td>0.0</td>
<td>0.180165</td>
<td>0.0</td>
<td>0.121395</td>
<td>0.69844</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0</td>
<td>0.0</td>
<td>0.178332</td>
<td>0.0</td>
<td>0.119651</td>
<td>0.702016</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>20</td>
<td>0.0625</td>
<td>0.125</td>
<td>0.25</td>
<td>0.0625</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>0^4</td>
<td>0.0592</td>
<td>0.1089</td>
<td>0.2373</td>
<td>0.0592</td>
<td>0.2373</td>
<td>0.2981</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.06021</td>
<td>0.12939</td>
<td>0.24991</td>
<td>0.06021</td>
<td>0.24991</td>
<td>0.25037</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.0630975</td>
<td>0.122252</td>
<td>0.248098</td>
<td>0.0630975</td>
<td>0.248098</td>
<td>0.251157</td>
<td></td>
</tr>
<tr>
<td>Id</td>
<td>22</td>
<td>0.125</td>
<td>0.25</td>
<td>0.25</td>
<td>0.125</td>
<td>0.25</td>
<td>0.25</td>
</tr>
<tr>
<td>0^4</td>
<td>0.11785</td>
<td>0.218</td>
<td>0.27315</td>
<td>0.11785</td>
<td>0.27315</td>
<td>0.27315</td>
<td></td>
</tr>
<tr>
<td>10^5</td>
<td>0.12429</td>
<td>0.24771</td>
<td>0.251855</td>
<td>0.12429</td>
<td>0.251855</td>
<td>0.251855</td>
<td></td>
</tr>
<tr>
<td>10^6</td>
<td>0.124293</td>
<td>0.247168</td>
<td>0.252124</td>
<td>0.124293</td>
<td>0.252124</td>
<td>0.252124</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.3: Estimated marginal genotype probabilities in pedigree with loops obtained by different chain lengths of the segregation indicator sampler and exact marginal probabilities for the individuals in with unknown genotype of the half-sib family in Fig. 3.8.
3.4 Performance of Segregation Indicator Samplers

<table>
<thead>
<tr>
<th>Pr(AA)</th>
<th>Pr(AB)</th>
<th>Pr(A0)</th>
<th>Pr(BB)</th>
<th>Pr(B0)</th>
<th>Pr(00)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 and 3</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>6, 9 and 10</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>15</td>
<td>0.5</td>
</tr>
<tr>
<td>16, 17, 18 and 19</td>
<td>1</td>
<td>21</td>
<td>1</td>
<td>23</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 3.4: Genotype probabilities in half-sib family pedigree with loops (Fig. 3.8): Table of individuals where sampled genotype probabilities were equal to the exact marginal probabilities.

3.4.4 Location Scores

To test the combination of the stochastic segregation indicator sampler at marker loci with deterministic Maximum Likelihood (ML) methods at a biallelic QTL, the initial motivation for the present thesis, phase information from segregation indicator sampling was used in deterministic ML to peel a bi-allelic QTL. Iterative peeling (Wang et al., 1996) adapted to accept phase information from the segregation indicator samplers was used to account for complex pedigrees. To obtain a test statistic for the QTL position, the expected likelihood conditional on marker phase information $E\{L(y|G)\}$ is calculated for a set of samples from the Markov chain of segregation indicators. A grid of location scores can be obtained by repeatedly calculating $E\{L(y|G)\}$ for a set of QTL positions.

The pedigree as depicted in fig 3.9 with four marker and a bi-allelic QTL with additive effect $f$ for allele $A$ and 0 for allele $a$ was simulated. The four marker alleles and the QTL alleles are equally likely. The marker are at positions 0.0M, 0.26M, 0.32M and 0.57M while the QTL is simulated at position 0.27M, the simulated residual variance was 0.5. Using the S&L-sampler at marker loci with deterministic iterative peeling for this pedigree, a grid of QTL location scores has been plotted using a grid density of 0.01M (Fig. 3.10). For each step in the grid, $10^3$ samples have been drawn after a burn-in period of $10^3$.
samples. The average computing time for the grid was 390 seconds on a 350 MHz INTEL Pentium II running LINUX 2.4. The Fig. 3.10 shows that the likelihood for the QTL position is highest at the position where the QTL was simulated although the likelihood level is indicating that only few information is present in the data.

Figure 3.9: Artificial pedigree with loops

Figure 3.10: Grid of QTL location scores: Simulated position of the QTL is at 0.27M
3.4.5 More Loci

To demonstrate the performance of the segregation indicator samplers in multi-allelic cases, consider the pedigree depicted in Fig. 3.11 with all individuals genotyped at two tightly linked loci 0.01 cM apart. Only the maternal meiosis of the terminal offspring are informative. There are 4 ordered possible genotypes for mother 10, but only the orders 1|12 and 2|21 (| indicates a haplotype) are likely to be sampled, due to the loci being tightly linked and 10 terminal offspring 100 to 110 receiving a haplotype |1 from mother 10. The two ordered genotypes are expected to be sampled with equal probability of virtually 0.5. The terminal offspring 100-110 are expected to get their |1 haplotype from the maternal grandmother or maternal grandfather with probability 0.5.

The results in Table 3.5 show that the S&L-sampler is locked in the starting configuration due to strong dependence between loci (horizontal dependence due to tight linkage) while the haplotype (HT) sampler and the cascading origin (CO) sampler perform equally well, although with quite different acceptance rates. The HT-sampler performs better in this example than the CO-sampler because the CO-sampler will frequently propose large steps incompatible to the data while the HT-sampler will almost every time sample descent graphs compatible to the genetically uniform offspring due to the tight linkage.
Table 3.5: Proportion of ordered genotypes for mother 10 in Fig. 3.11 (10^5 rounds for each sampler).

<table>
<thead>
<tr>
<th></th>
<th>ordered genotype</th>
<th>acceptance rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sobel &amp; Lange (S&amp;L)</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>haplotype sampler (HT)</td>
<td>0.5005</td>
<td>0.4995</td>
</tr>
<tr>
<td>cascading origin (CO)</td>
<td>0.4973</td>
<td>0.5027</td>
</tr>
<tr>
<td>ESIP</td>
<td>0.4977</td>
<td>0.5023</td>
</tr>
</tbody>
</table>

For comparison, the results from the an ESIP sampler (c.f. section 2.2.3 on page 38) are included. The acceptance rate for the ESIP sampler is 1.0 since the pedigree is simple and can thus be peeled exactly.

Assume again the pedigree in Fig. 3.11 but with an additional terminal offspring 111 having the unordered genotypes 1,3 at the first and 2,3 at the second locus (Fig. 3.12). Following the same reasoning as before, offspring 111 is now likely a recombinant.

This extension of the pedigree has no effect for the SL-sampler, i.e. the chain is locked in the starting configuration due to the same reasons as above (results in Table 3.6). However, for this extended pedigree with offspring 111 being a likely recombinant, the HT-sampler does not reach the true posterior distribution, as it tends to reject all recombinants between
3.4 Performance of Segregation Indicator Samplers

| sampler                           | ordered genotype $1||2$ | ordered genotype $2||1$ | acceptance rate |
|----------------------------------|-------------------------|-------------------------|-----------------|
| Sobel & Lange (S&L)              | 1.0                     | 0.0                     | 0.084           |
| haplotype sampler (HT)           | 0.4025                  | 0.5975                  | 0.369           |
| cascading origin (CO)            | 0.5061                  | 0.4939                  | 0.274           |
| ESIP                             | 0.4977                  | 0.5023                  | 1.0             |

Table 3.6: Proportion of ordered genotypes for mother 10 in Fig. 3.12 (10^5 rounds for each sampler).

tightly linked loci. Acceptance rates stay as in the previous example, except that the re-combinant offspring 111 lowers it for the HT-sampler. Note that all samplers demonstrated here are theoretically irreducible. Thus it is practical irreducibility encountered in these examples caused by horizontal dependence. More complex pedigree structures and genetic models, when a QTL is sampled also, will likely cause the SL- and the HT-sampler to fail in practice. In these examples, the CO-sampler performed well, although considerably less efficient than the ESIP-sampler. Therefore, when only a limited number of markers is considered (e.g. interval mapping), then the ESIP-sampler performs best since it samples the whole pedigree jointly conditional on the data. If more loci need to be accounted for, jointly sampling all genotypes for the whole pedigree becomes inefficient. Combining the SL-, HT- and the CO-sampler is then expected to be more efficient. Such a combined sampling strategy is implemented in the preliminary version of MATVEC with the number of SL-, HT- and CO rounds to be determined by the user.

### 3.4.6 Large Pedigree

The pedigree in Fig. 3.13 was used to indicate that three segregation indicator samplers can be applied to large and complex pedigrees common in animal breeding. Recall that daughter- and grand-daughter-designs are in fact complex pedigrees where relationships between certain individuals are disregarded for statistical convenience. Analysis in such designs is usually based on information which QTL allele is passed from a parent to its
offspring, i.e. on probabilities of descent for QTL allele (PDQ’s). The pedigree consists of 325 members, 16 of them, shaded grey in Fig. 3.13, with known genotype at two markers flanking a QTL. The marker have four and five equally frequent alleles, respectively. The recombination fraction between marker and the QTL is 0.1 for both QTL - marker intervals. The initial configuration for the descent graph sampler was computed by an adapted version of ESIP (Fernández et al., 2001).

PDQ’s were estimated using a sequence of 10 rounds of the S&L-, HT- and CO-sampler each, the total chain length was $10^6$. Computing time on a 1 GHz INTEL PC running LINUX 2.4 was around 19’000 seconds. The mean PDQ’s for the females of the last and the second last generation were 0.499998714 and 0.500001286 for the first marker and 0.50046955 and 0.49953085 for the second marker. The PDQ’s for the male individuals in generation 4 (Table 3.7) show that the Markov chain was not locked in a specific configuration.
### Table 3.7: PDQ’s of male offspring in generation 4 of pedigree in Fig. 3.13.

<table>
<thead>
<tr>
<th>Individual</th>
<th>$\Pr(QTL)$ descending from maternal granddam</th>
<th>$\Pr(QTL)$ descending from paternal granddam</th>
<th>$\Pr(QTL)$ descending from maternal grandsire</th>
<th>$\Pr(QTL)$ descending from paternal grandsire</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>0.031287</td>
<td>0.968713</td>
<td>0.03616</td>
<td>0.96384</td>
</tr>
<tr>
<td>2000</td>
<td>0.034896</td>
<td>0.965104</td>
<td>0.967256</td>
<td>0.032744</td>
</tr>
<tr>
<td>4000</td>
<td>0.065733</td>
<td>0.934267</td>
<td>0.837308</td>
<td>0.162692</td>
</tr>
<tr>
<td>5000</td>
<td>0.038874</td>
<td>0.961126</td>
<td>0.619088</td>
<td>0.380912</td>
</tr>
</tbody>
</table>
3 Combining Deterministic Calculation and Stochastic Simulation
4.1 Sampling Genotypes

Initially, sampling genotypes by the Gibbs Sampler was thought to be a major breakthrough in QTL mapping. However, it was soon realized that for loci with more than two alleles, sampling single genotypes by a Gibbs sampler does not guarantee an irreducible Markov chain (c.f. section 2.2.1). The example of a nuclear family with unknown parents and two offspring with genotype AB and CC, respectively has been frequently used to illustrate this (Fig. 2.12). Several authors claim that the Markov chain is irreducible if at least one parent has observed marker genotypes, a situation frequently encountered in livestock pedigrees where DNA on male parents is usually available. The counter example in Fig. 2.14 demonstrates that this statement is not true in general (Cannings and Sheehan, 2002).

Updating blocks of variables (genotypes) jointly may provide an irreducible Markov chain and thus overcome the problems illustrated by Fig. 2.14. However, there is no general algorithm available to construct blocks of genotypes such that irreducibility is guaranteed in general pedigrees, except when the block consists of the entire pedigree. The latter is the basis of the approach taken by Fernández et al. (2001; 2002) (c.f. section 2.2.3). For
simple pedigrees, the entire pedigree is peeled exactly using the Elston-Stewart algorithm, then ordered genotypes are sampled by reverse peeling (Heath, 1998). If a pedigree is complex and hence cannot be peeled exactly, they use a modified pedigree to draw joint genotypic realizations from. Genotypes are peeled exactly until loops make peeling too inefficient. Then, the remaining loops are cut and the pedigree is extended at the cuts. Finally, samples are generated from the modified pedigree, and these are accepted with the probability given by the Metropolis-Hastings ratio. Since the proposal distribution (i.e., the modified pedigree) is very close to the true pedigree, an independence sampler is used. Fernández et al. call this the ESIP-sampler, combining the Elston-Stewart algorithm and iterative peeling. In Fernández et al. (2002) they show that the Markov chain generated by the ESIP-sampler is irreducible and aperiodic. Extension of the ESIP-sampler to multiple loci is straightforward, implying that peeling needs to be performed over several loci jointly. However, as the sampler relies on peeling, efficiency decreases exponentially with increasing number of loci considered in the model. For multi-point linkage studies using flanking markers, the ESIP-sampler does provide a fast mixing, efficient sampler. The ESIP sampler has been extended to sample genotypes at multiple loci by a sequential imputation-type approach. The genotypes at locus \( t \) are samples conditional on data and on the genotypes of previously sampled adjacent genotypes \( t - 1 \). A publication regarding this approach is in preparation (Fernando, 2003, pers. comm.).

### 4.2 Sampling Allelic Origin

Thompson (1994) and Sobel and Lange (1996) distinguish two types of genetic variables in a pedigree: 1) the grand maternal or grand paternal origin of each non-founder allele, which have been called segregation indicators, and 2) the ordered genotype of each individual. Individuals with observed genotypes put restrictions on the size of the state spaces for these variables. The state space for the latter, which is the set of all ordered genotypes compatible with the observed data, is usually substantially larger. The information that is useful for
4.2 Sampling Allelic Origin

making inferences on the linkage between loci is contained entirely in the segregation indicators. In the algorithms proposed by Thompson (1994) and Sobel and Lange (1996), segregation indicators for non-founders were sampled without sampling the genotypes of the founders. To do so, they first computed the joint probability of the founder genotypes and the non-founder segregation indicators; then, the marginal probability for the segregation indicators was obtained by deterministically taking the sum of these probabilities over all the possible founder genotypes that are compatible with the observed data. Segregation indicators were sampled from this marginal distribution.

In recent applications to livestock and plant pedigrees, in addition to segregation indicators, the genotypes of founders were also sampled (Jansen et al., 1998; Bink and Van Arendonk, 1999; Yi and Xu, 2001). The size of the state space for segregation indicators and for ordered genotypes depend on the amount of unobserved marker genotypes and the size of the pedigree. There is no simple rule to identify which state space is smaller. Note that the term ‘order’ refers to the maternal or paternal origin of an allele, whereas the term ‘phase’ refers to the placement of alleles onto haplotypes across several loci without specifying the maternal or paternal origin of the haplotype. Thus for multiple loci, order includes all information about phase, whereas when a only single locus is considered the property phase does not apply. In the recent applications listed above performing simulations on the state space of segregation indicators, the Gibbs sampler was used. Unfortunately, the Gibbs sampler does not guarantee an irreducible Markov chain in such situations as shown in Fig. 4.1.

Therefore, the Gibbs sampler may fail even for sampling the order of known genotypes, when segregation indicators are sampled (Note that for the pedigree in Fig. 4.1 genotypes of all individuals are known). It is due to vertical dependence, i.e. dependence between individuals’ loci. Therefore, simulations on the state space of segregation indicators, although initially appealing due to the potentially smaller state space compared with the space of all genotypes revealed that common single site Gibbs sampling is not guaranteed
Figure 4.1: Simple pedigree demonstrating vertical dependence. Applying a Gibbs sampler results in a reducible Markov chain with non-communicating sets I. and II. I. corresponds to the ordered genotype of \([1,2]\) for the middle offspring, II. to \([2,1]\). Both are equally likely given the data.

to result in an irreducible Markov chain. Hence, it cannot be applied in general. A further example showing the failure of the Gibbs sampler to sample (ordered) genotypes was demonstrated in Fig. 3.1 and 3.2.

Sobel and Lange (1996) do not simulate from the full conditional distribution of a segregation indicator given all remaining segregation indicators and the data like in the Gibbs sampler. They use a Metropolis sampler to simulate from the conditional distribution of segregation indicators only. Their sampler will mix poorly due to only small changes being compatible with the data. To improve mixing, they update segregation indicators for certain blocks of segregation indicators jointly with the blocks chosen based on genetic relationship between individuals (full- or halfsibs). Although this resolves the problem inherent in Fig. 4.1, it will not in general guarantee an irreducible Markov chain (c.f. section 2.2.6). To achieve this, they use a random number of updating steps to propose a new realization of the Markov chain. The random number is taken from a geometric distribution with mean 2. Such multiple updating steps allow to step through illegal transmission patterns, revealing a theoretically irreducible Markov chain.

The example of a single nuclear family in section 3.4.1 and of a large half-sib family in section 3.4.2 show the properties of the Sobel & Lange segregation indicator (S&L-) sampler.
4.2 Sampling Allelic Origin

Figure 4.2: Example of vertical dependence overcome by blocking within families: Applying the $T_1$-rule allows to switch in one step between both possible ordered genotypes for the $\{1,2\}$ individuals whereas applying a single $T_0$-rule does not.

Originally, the potentially smaller state space for descent graph sampling was thought to reveal good mixing properties. Although the grouping strategies by Sobel and Lange discussed in section 2.2.7 are well adapted to small pedigrees with designed structures such as full- or half-sib families like in Fig. 4.2, they may fail in more complex pedigree structures. Large numbers of offspring with observed genotypes in half-sib families, often encountered in animal breeding, may lead to slow mixing Markov chains. This can be demonstrated by the example shown in Fig. 4.3. To move from the first descent graph with the genotype $\{A,A\}$ for the middle founder to the second descent graph with the genotypes $\{A,A\}$ or $\{A,a\}$ for the middle founder requires the sequence of two source switches. The probability for such a sequence in this small pedigree is the probability that the appropriate allelic node is selected times the probability for a second transition times the probability for the appropriate second allelic node of the founder to be selected, $\Pr(n_i) \Pr(t) \Pr(n_j) = \frac{1}{14} \cdot \frac{1}{2} \cdot \frac{1}{14}$ (assuming a geometric distribution with mean 2 for $t$ and applying only $T_0$-transitions). This likelihood will become virtually nil in large populations.

As soon as vertical dependence, i.e. dependence between different individuals, becomes an issue, the S&L-sampler shows low efficiency in general as demonstrated in the example shown in Fig. 3.3 and Fig. 4.3. In the half-sib family example shown in Fig. 3.3, the initial descent graph becomes important for the mixing of the chain, i.e. communication between different descent graphs. For this pedigree, estimates based on initial configuration 1 (Fig. 3.4) and 2 (Fig. 3.5) converge slowly to the true marginal probabilities (Table 3.2).
Figure 4.3: Example of vertical dependence in a small pedigree where the grouping strategies by Sobel and Lange are inefficient: Moving between the two descent graphs requires a sequence of two source switches ($T_0$-rule applied to the bold arrows).

Estimates based on initial configuration 3 (Fig. 3.6) and 4 (Fig. 3.7) do virtually not converge to the true marginal probability for any of the individuals with unknown genotypes due to vertical dependence. Note however that the sampler is (theoretically) irreducible. Fernández et al. (2002) showed that the ESIP-sampler applied to the same example is far more efficient, she obtained convergence to the true marginal probabilities with only 10'000 samples. This is due to the ESIP-sampler considering the whole pedigree as a single block and thus eliminating any vertical dependence problems. However, efficiency for the ESIP-sampler will considerably drop in multi-locus analysis as will be discussed below.

The application of the S&L-sampler to pedigree with loops is shown in the ABO-blood type pedigree by Sheehan and Thomas (1993) shown in section 3.4.3 and Fig. 3.8. Convergence to the true marginal genotype frequencies for the individuals with unknown genotypes is rapidly achieved (Table 3.3 and 3.4). Therefore, in contrary to deterministic Maximum Likelihood methods, the S&L-sampler is only sensitive to the pedigree structure when it generates strong vertical dependence.

Fernández et al. (2002) compared the performance of the S&L-sampler to the ESIP-sampler. Using the same number of samples, the ESIP-sampler showed better convergence.
4.2 Sampling Allelic Origin

to the true marginal probabilities. In this example, ESIP is more efficient and accurate. This was mainly due to the fact that although their example pedigree was complex, it still could be exactly peeled. This can not be done if the cutset size becomes too large in more complex pedigrees.

To improve the mixing of a sampler, it is necessary that large steps have a reasonable high acceptance ratio. However, large steps compatible with the observed data are rare if sampling is not conditional on data. Du and Hoeschele (2002) proposed to update large blocks jointly conditional on observed data, thus provide the desired high acceptance ratio for large steps. To improve efficiency, they first split the pedigree in a main pedigree with segregation indicators which are dependent of each other and a side pedigree with all segregation indicators independent of any others. These segregation indicators can be sampled independently, thus restricting the state space of the simulation of the main pedigree. In a second step, the segregation indicators of the main pedigree are grouped together (meiosis grouping). First, all fixed segregation indicators, i.e. which can only take a single value to be compatible with observed data are identified by genotype elimination (Du and Hoeschele, 2000) and grouped. For the remaining segregation indicators they implemented four different grouping strategies. Strategy 1 and 2 group based on genetic relationship all segregation indicators of full-sib or all indicators derived from a single parent as in Sobel and Lange ($T_2$ and $T_1$ rule). In contrast, strategies 3 and 4 are based on observed marker data. In strategy 3, segregation indicators which restrict the grandparental origin of other segregation indicators are grouped and in strategy 4 segregation indicators which force changes of the allelic state of other segregation indicators.

Du and Hoeschele further improve the acceptance ratio for the sampler by updating groups sequentially. The resulting incomplete descent graphs are also evaluated sequentially conditional on observed data. Additionally, they presented a second sampling strategy where groups of segregation indicators are jointly updated conditional on the data in the whole pedigree.
Du and Hoeschele show that their sampling strategies reveal irreducible Markov chains either by the random and unlimited size of the jointly update groups or by design (sequential update of all groups). Considering multiple loci independently will generate a poorly mixing sampler for tightly linked loci. The grouping strategies of Du and Hoeschele (2002) must thus be extended to include multiple loci.

Until now, the performance of different approaches to sample allelic origin has been discussed with respect to a single locus only. Issues to be addressed where vertical dependence and acceptance rates (efficiency). The latter was especially low for the S&L-sampler compared to the ESIP-sampler as the S&L-sampler does not sample its proposals conditional on data as the ESIP-sampler does.

If more than a single locus is considered, then the problem of horizontal dependence arises. Sampling multiple loci individually by the S&L-sampler, reveals poor mixing for tightly linked loci as neither linkage nor interference between loci is considered when a new descent graph is proposed. This will lead to a very low acceptance rate, i.e. low efficiency of the S&L-sampler. Not only that the proposals are sampled unconditional on data, they are generated as if the loci would be independent. To overcome these limitations, a haplotype sampler using Metropolis-Hastings acceptance ratios (and not the Gibbs sampler as in Thompson and Heath (1997)) was derived. In section 2.2.9 it was demonstrated that such a sampler will tend to ‘loose’ all recombinations between tightly linked loci. The reason for this is that an appropriate mapping function for tightly linked loci will virtually always generate non-recombinant haplotypes. This will lead to slow mixing in situations when rare recombinants between such tightly linked loci are present in the data due to the proposed graphs being rejected as incompatible with the data (c.f. section 3.4.5). However, an advantage of the haplotype sampler is, that for tightly linked loci, it will move away rapidly from very unlikely descent graphs. The shortcomings of the haplotype sampler lead to the derivation of the Cascading Origin-sampler (CO-sampler). The CO-sampler is an approach to sample segregation indicators of haplotypes jointly. As in the haplotype sampler, segregation indicator are updated according to the principles described in Sobel
4.2 Sampling Allelic Origin

and Lange (1996). According to a mapping function, recombination probability is calculated between the updated locus \(i - 1\) and the adjacent locus \(i\). With probability \(1 - \theta\) the same update as on segregation indicator \(i - 1\) is performed on segregation indicator \(i\). With probability \(\theta\) the alternative action as on segregation indicator \(i - 1\) is performed on segregation indicator \(i\). This process is also used to sample segregation indicators at locus \(i - 2\), and the process is continued to the left and right until all loci are sampled. Unlike the haplotype sampler, this approach will not tend to loose recombination events between tightly linked loci in its proposals.

To combine the advantages of all segregation indicator samplers in multilocus situations, i.e. original S&L, haplotype and CO sampling, all three samplers were combined when they were implemented in MATVEC: First \(k_1\) rounds of original S&L sampling, then \(k_2\) rounds of haplotype sampling and finally \(k_3\) rounds of CO sampling are executed with each \(k_1, k_2\) and \(k_3\) being non-negative integer. The proportions \(k\) have to be pre-specified and cannot be changed according to the mixing of the chain during program run to maintain a symmetric proposal distribution without violating the major property of Markov chains, namely that state \(t^*\) is sampled only conditional on state \(t_{-1}\).

As deterministic maximum likelihood is a well established approach in segregation analysis, the main emphasis of this thesis was on deriving appropriate samplers for marker inheritance patterns. Such marker inheritance patterns can be used to estimate the covariance matrix at a marked QTL or as phase information to conditionally peel a QTL. The latter was shown in section 3.4.4 through a grid of location scores. Of course, other approaches to maximize over QTL positions can be derived. The feasibility to estimate the probability of genes shared identical by descent in general pedigrees will boost not only variance component based QTL-mapping approaches based on grand-daughter or daughter designs, it will have even more profound influence on approaches for marker assisted selection (MAS) as it allows to calculate the covariance matrix at marked QTL, the basis of marker assisted best linear unbiased prediction (BLUP).


Curriculum vitae

Family name: Schelling

First Name: Matthias Jakob

Date of Birth: May 09, 1970

Citizenship: St. Gall and Berneck, Switzerland

Education

1977 - 1983 Primary school in Altnau (TG), Switzerland

1983 - 1985 Secondary school in Altnau (TG), Switzerland

1985 - 1990 Gymnasium type B, Kantonsschule Kreuzlingen (TG), Switzerland

1990 - 1996 University studies in Agriculture, Swiss Federal Institute of Technology ETH, Zurich, Switzerland

1996 Diploma in Agriculture with emphasis on Genetics and Animal Production, Swiss Federal Institute of Technology ETH, Zurich, Switzerland

Certificate in Didactics, Swiss Federal Institute of Technology ETH, Zurich, Switzerland

Professional activities

1996 - 2002 Research assistant at the Institute of Animal Science, Swiss Federal Institute of Technology ETH, Zurich

2002 - Geneticist and section leader, genetic evaluation, Swiss Red & White Cattle Breeder’s Federation