

ROBUST METHODS FOR ANALYSING QUANTITATIVE
TRAIT LOCI

by

Nuovella Williams

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Doctor of Philosophy
in Statistics.

Victoria University of Wellington

2006

Abstract

The advent of new technology for extracting genetic information from tissue samples has increased the availability of suitable data for finding genes controlling complex traits in plants, animals and humans. Quantitative trait locus (QTL) analysis relies on statistical methods to interpret genetic data in the presence of phenotype data and possibly other factors such as environmental factors. The goal is to both detect the presence of QTL with significant effects on trait value as well as to estimate their locations on the genome relative to those of known markers.

This thesis reviews commonly used statistical techniques for QTL mapping in experimental populations. Regression and likelihood methods are discussed. The mixture-modelling approach to QTL mapping is explored in some detail. This thesis presents new matrix formulas for exact and convenient calculation of both the Observed and Fisher information matrices in the context of Multinomial mixtures of Univariate Normal distributions. An extension to Composite Interval mapping is proposed, together with a hypothesis testing strategy which is robust enough to detect existing QTL in the presence of slight deviations from model assumptions while reducing false detections.

Acknowledgements

This work was funded by a Commonwealth Scholarship from the New Zealand Vice Chancellor's Committee. I also acknowledge the help of the Training Division of the Government of Montserrat in facilitating access to this research opportunity.

Many thanks to my supervisors Dr. Richard Arnold and Dr. Ross Renner for their help and encouragement throughout this research project. I am also grateful to the administrative and computing staff of the School of Mathematics, Statistics and Computer Science (Victoria University of Wellington) for their support.

VUW, New Zealand
December, 2006

Nuovella Williams

Table of Contents

Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Tables	vii
List of Figures	ix
1 Introduction	1
1.1 Successes	5
1.2 Usefulness and Verifiability	6
1.3 The Challenges of QTL Mapping	8
1.4 Review of the model-development literature	12
1.5 Contribution of this Thesis	18
1.6 Thesis Layout	18
2 Linkage, Breeding Designs and Genetic Effects	20
2.1 Linkage and Recombination Fractions	20
2.2 Breeding designs	23
2.3 Genetic effects	27
2.3.1 Additive, dominance and epistatic effects	27
2.3.2 Harmonized definitions of genetic effects	35
2.3.3 Partitioning the genetic variance	37
2.3.4 Number of genetic effects in a full linear regression model . . .	39
3 The Inherent Mixture	41
3.1 Statistical Exploration of Line-Cross data	41

3.2	From marker to QTL	46
4	Regression Methods	53
4.1	Multiple Regression with Contrasts	54
4.1.1	Models, contrasts and implications	54
4.1.2	Sums of Squares and Hypothesis Tests	60
4.1.3	Inferring QTL from marker regression	63
4.1.4	Choice of Contrasts	68
4.2	An example based on single-marker regression	71
5	A Robust Interval Mapping Procedure	77
5.1	The Model Specification for RIM1	79
5.1.1	Genotypic content of the backcross population at the loci under study	79
5.1.2	Relating genotypic content to trait value	81
5.1.3	The model matrix and likelihood function for a sample	90
5.2	Maximum Likelihood Analysis	93
5.2.1	Maximization Procedure	93
5.2.2	The conditional observed information matrix	102
5.2.3	The Fisher information matrix	110
5.3	Hypothesis testing	111
5.4	Computational Issues	124
5.4.1	Selecting starting points for the EM Algorithm	124
5.4.2	Stopping rules	129
5.4.3	Adjustments to RIM1	130
5.4.4	Reduced Models for fitting fewer than three QTL	131
5.4.5	The possibility of a singular information matrix	132
5.4.6	Programming environment	133
6	Information Matrix Derivations	135
6.1	The Complete-Data Conditional Information	136
6.2	Notation and Useful Matrix Identities	138
6.2.1	Notation	138
6.2.2	General Matrix Identities	139
6.2.3	Matrix Identities that are Specific to our Problem	140
6.3	Conditional Expectations of Products of the Estimated Category Identities	144
6.4	Conditional Expectations of Outer Products of the Score Vectors	148
6.5	The Conditional Observed Information Matrix	171

6.6	Intermediate Results Involving Integrals	174
6.7	The Fisher Information Matrix	179
7	Simulations and Results	189
7.1	The Single-QTL Situation	190
7.1.1	Quality of the MLEs of QTL effect and position	197
7.1.2	Performance of the Fisher information matrix	209
7.1.3	Hypothesis testing	220
7.2	The Multi-QTL Situation	226
8	Other Breeding Designs and Real Data Applications	230
8.1	Including interactions between QTL	230
8.2	Application to Other Inbred Designs	231
8.2.1	Application to the F2	232
8.2.2	Designs involving loci with more than two alleles	235
8.3	Applications to real data	236
8.3.1	Real F2 Application	237
8.3.2	Real Backcross Application	240
8.4	Overview	245
9	Summary and Conclusions	246
A	Constructing an Orthogonal Contrast Matrix	251
B	Programs and Code	254
B.1	R code for parameter estimation in RIM1 and its sub-models	255
B.2	Utility functions for QTL analysis (R Code)	307
B.3	Examples of using the utility functions with QTL Cartographer	316
B.4	R code to implement the information matrix formulas for RIM1 and its sub-models	322
B.5	Using the RIM1 functions in batch mode - an example	351
B.6	Permutation tests with RIM1	353
B.7	Using the RIM1 functions with the Horvat and Medrano mouse data	355
	References	359

List of Tables

3.1	Some population properties of line cross designs	43
3.2	Some sample properties of line cross designs	44
4.1	One-way ANOVA table.	62
4.2	Contrasts to extract genotypic effects for a backcross model	69
4.3	Contrasts to extract genotypic effects for an F2 model	70
5.1	QTL genotypes and their indices in a B ₁ -backcross model with loci in the order $L-M-Q-N-R$	80
5.2	Calculation of $P(x_L x_K, x_M)$ for the B1 Backcross	86
5.3	Calculation of $P(x_Q x_M, x_N)$ for the B1 Backcross	86
5.4	Calculation of $P(x_R x_N, x_O)$ for the B1 Backcross	86
5.5	Conditional genotype probabilities, w_{ik} , for the B1 Backcross	87
6.1	Selected conditional expectations involving products of estimated cat- egory identities	144
6.2	List of propositions that provide formulae for calculating the condi- tional expectation of the outer product of the score vector	148
6.3	List of integrals used for calculating of the Fisher information matrix.	174
7.1	An Example of raw output from our RIM1 implementation	193
7.2	Summarising the output of RIM1	196
7.3	Summarising QTL Cartographer output for CIM	196

7.4	Simulated Single-QTL Case: Interval c2m7-c2m8; comparison of estimated standard errors (SD) of \hat{b}_Q	210
7.5	Standard error of estimated QTL effect from RIM1 on replicates . . .	211
7.6	Standard error of estimated QTL effect from CIM on replicates . . .	211
7.7	Simulated Single-QTL Case: Interval c2m7-c2m8; comparison of estimated standard errors (SD) of \hat{p}_{Q2}	213
7.8	MLE of QTL location and its estimated standard error from RIM1 on replicates	214
7.9	MLE of QTL location and its estimated standard error from CIM on replicates	215
7.10	RIM1 on bootstraps: standard error of estimated QTL effect.	219
7.11	RIM1 on bootstraps: standard error of estimated QTL location. . . .	219
7.12	Percent of times p-value < 0.001 for different tests applied to Simulated backcross data; single QTL, sample size 2000	221
7.13	Percent of times p-value < 0.001 for ten testing methods applied to Simulated backcross data; single QTL, sample sizes 500 and 125 . . .	222
7.14	Multi-QTL case: QTL locations and effects used for simulations. . . .	226
7.15	Percent of times p-value < 0.001 for nine testing methods applied to multi-QTL situation; simulated backcross data	228
8.1	Marginal genotype probabilities in the F2 for three loci	233
8.2	Conditional QTL genotypic probabilities in the F2	233
8.3	Results of applying RIM1 to a real F2 dataset	239
8.4	Results of applying RIM1 to a real backcross dataset	244
8.5	Results of bootstrapping a real backcross dataset	245
B.1	List of functions used for model fitting	255
B.2	List of utility functions	307
B.3	List of information matrix functions	322

List of Figures

2.1	Definitions of backcross and F2 progeny	25
2.2	Definitions of Second Backcross and Doubled Haploid lines	25
5.1	Model genetic map	79
5.2	Plots of p_{Q1} versus r_{MQ} and $\min(p_{Q2})$ versus r_{MN}	88
5.3	Plot of p_{Q2} versus r_{MQ}	89
5.4	Grid for selecting starting values for the EM Algorithm	125
7.1	Single-QTL, genetic map on which simulations were based	191
7.2	Scatter plots of \hat{r}_{MQ} from CIM based on simulated samples of sizes 125, 500 and 2000	198
7.3	Scatter plots of \hat{r}_{MQ} from RIM1 based on simulated samples of sizes 125, 500 and 2000	200
7.4	Box plots of \hat{b}_Q and from CIM and CIM-QTLcart, based on sim- ulated samples with a single QTL and sample size 2000	202
7.5	Box plots of \hat{b}_Q and $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ from RIM1 based on simulated samples with a single QTL and sample size 2000	203
7.6	Box plots illustrating the ability of RIM1 to detect QTL to the left and the right of a testing interval	204
7.7	Box plots of $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ from CIM and CIM-QTLcart, based on simulated samples with a single QTL and sample size 2000	206
7.8	Box plots of \hat{b}_Q and $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ from RIM1 based on simulated samples with a single QTL and sample size 500	207

7.9	Box plots of \hat{b}_Q and $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ from RIM1 based on simulated samples with a single QTL and sample size 125	208
7.10	Box plots showing distributions of \hat{p}_{Q2} from RIM1	214
7.11	Box plots showing distributions of \hat{p}_{Q2} from CIM	215
7.12	Box plots of the estimates \hat{p}_{L2} , \hat{p}_{Q2} and \hat{p}_{R2} obtained from RIM1 . . .	216
7.13	Using Permutations to remove a QTL effect	225
7.14	Multi-QTL, genetic map on which simulations were based	227
8.1	Horvat and Medrano mouse map	237

Chapter 1

Introduction

A trait is a quantitative or qualitative characteristic of an individual that is observable and that is used to define a phenotype or character of interest. Phenotypes are generally classified according to the type of trait values (discrete/qualitative, continuous/quantitative) used to define them, or according to the mode of inheritance (Mendelian inheritance, complex inheritance) which is hypothesized by the analyst. Gelderman (1975) coined the phrase Quantitative trait locus (or loci) (QTL) to mean a gene (or genes) controlling a quantitative character.

QTL detection techniques use statistical tools to determine if genes significantly affecting the expression of a trait exist within a given search region on a particular chromosome. QTL detection aims to estimate genetic effects and mean trait values within genotype groupings.

QTL mapping goes further, by using more specialized statistical tools to determine approximate QTL positions relative to those of other genes, called markers, whose chromosomal locations are known. QTL mapping estimates the genetic distance between a QTL and a marker. The genetic distance is the number of crossovers or recombinations that occur between the two loci during meiosis, whereas the physical distance between them is the number of nucleotide pairs (base pairs) between the loci.

Genetic distance is measured in Morgans or centiMorgans (cM) , where a Morgan is the distance over which one recombination event is expected to occur per generation, and one centiMorgan is equal to 0.01 Morgans.

The relationship between genetic distance and physical distance can vary at different points along a chromosome and it varies from species to species. In humans, 1cM is approximately equal to one million base pairs. QTL analysis is motivated by the need to understand the mechanisms governing one or more quantitative traits, to find the genes involved and to understand their cellular functions.

In studies of agronomically important plants and animals, the traits which captivate the attention of researchers are those which affect productivity. Consider for example, the QTL analysis of soybean seed protein and seed oil by Chung *et al.* (2003) where the trait-values were assayed using near-infrared reflectance spectroscopy. This method enabled the protein content and seed oil to be quantified by weight (in grams per kilogram of dried meal), so that the values were suitable for quantitative data analysis.

Price *et al.* (1997) analysed genetic contributions to drought resistance in up-land rice by searching for associations between genetic markers and two shoot-related mechanisms, stomal closure and leaf rolling, which are evident in rice and which reduce transpirational water loss. They measured stomal closure by using a special instrument called a porometer and then they created three trait assessments from the porometer readings: stomal resistance before excision, time taken after excision to reach the fastest rate of stomal closure, and a score of the rate of stomal closure from one to four (slowest to fastest), based on visual assessment of plots of stomal resistance against time. They measured leaf rolling by the time (in minutes) taken for a young fully-expanded leaf to completely roll up after it was cut from the plant and placed on a flat bench. These traits (along with the corresponding genetic data) were separately analysed in order to search for QTL.

Another example is the QTL mapping study carried out by Spelman *et al.* (1999), involving New Zealand dairy cattle. They examined 17 non-production traits including traits such as adaptability to milking, shed temperament, stature, rump width, rump angle, live weight, udder support, teat placement and the farmer's overall opinion of each cow. The 17 traits were subjectively scored on a 9-point scale, where one and nine represented biological extremes, so we may say that these were pseudo-quantitative values.

In Human genetics and related studies the traits of interest are generally those associated with health and fitness or with disease susceptibility. For example, in order to conduct a genome-wide search for QTL underlying asthma, Xu *et al.* (2001) recorded several traits from individuals in a sample of 533 Chinese families. They studied nine asthma-related phenotypes including forced expiratory volume in one second, airway responsiveness to bronchoconstrictors and bronchodilators, serum total immunoglobulin E (IgE), serum-specific immunoglobulin E, eosinophil count in peripheral blood and skin-prick tests to three different allergens. The paper by Xu *et al.* (2001) gives very good detail on exactly how each phenotype was measured.

Animal models are often used to study the genetics of some diseases that affect human populations. Animal models (usually mouse models) have the advantage that they allow a researcher to implement controlled environments for trait development. In laboratory mice, traits may be induced by chemicals, by diet, by other environmental determinants or by genes.

The use of laboratory animals allows controlled breeding designs to be implemented so that an experimenter can limit the amount of genetic variation that occurs within the population. Also, very large sample sizes can often be obtained. Moreover, certain trait assay methods which cannot be applied to human samples can be used when working with animal models. For example, in order to study the genetics of two risk factors (lipoprotein levels and obesity) associated with coronary artery disease,

Warden *et al.* (1993) used a mouse model. Warden *et al.* (1993), measured several traits related to obesity, including body weight, body mass index, percent body fat. Some animals were sacrificed and dissected to obtain the weights (in grams) of three intra-abdominal fat pads as additional measures of obesity.

The examples above hint at the variety of traits and trait assessment schemes which are used in genetic association studies. Notice that trait assay can be carried out by methods that are as diverse as the use of precision instrumentation and the use of (sometimes *ad hoc*) subjective classification. It is not surprising that the chosen trait and its assay method can affect the choice of QTL detection method, the number and type of QTL detected and the ease of QTL detection. Solving the problem of trait assay is a huge challenge for experimenters. The choice of trait evaluation method may depend on the financial resources available, the available technology, the amenability of the species under study to a particular assay method, and may be governed by ethical and practical constraints.

Frankel (1995) gives a good illustration of the importance of trait definition in his discussion of a case where one disease assay criterion allowed the detection of a single QTL, but when a more accurate assay method was developed, two QTL were found. As suggested by Frankel (1995), the best policy is to assess several aspects of the phenotype, and to perform QTL analysis using the data from each aspect that has been evaluated.

Some researchers have proposed methods of simultaneously using several traits to search for QTL controlling them all (see Jiang and Zeng, 1995; Corander and Sillanpää, 2002). However, the implications of multiple-trait analysis are not well understood and consequently the most popular approaches to QTL mapping use single-trait analysis techniques.

This thesis looks at the statistical methods that are suitable for analysing *any* single continuous trait. We will only require that our trait of interest is continuous or

quantitative and that our assumptions about its distribution are reasonably justifiable within the sampled population. The underlying question will be: suppose that we have observed a continuous trait and that it has multiple genetic determinants, then how do we find the genes which control it and how do we separate pooled genetic effects?

1.1 Successes

QTL analysis studies have allowed successful detection of QTL associated with various traits in different species. Consider the QTL analysis studies cited in the previous section. Price *et al.* (1997) found one QTL for slow leaf rolling on chromosome 1 of the Bala rice variety. They also found two QTL for stomal closure: one located on chromosomes 3 and one located on chromosome 7 of the Bala rice genome. Spelman *et al.* (1999) discovered a QTL for stature on bovine chromosome 14 in New Zealand Dairy cattle but no QTL was found to be associated with the other 16 traits that they studied. Chung *et al.* (2003) detected a QTL for protein yield in soybean. Xu *et al.* (2001) found a very significant QTL for Asthma on human chromosome 2 and evidence for six other QTL of lesser effect.

Hundreds of other reported QTL detections may be found in the literature. Some of these results are available in on-line databases. For example, many QTL mapping results for rats, mice and humans are available from the Rat Genome Database (RGD), Rat Genome Database Web Site, Medical College of Wisconsin, Milwaukee, Wisconsin. World Wide Web (URL: <http://rgd.mcw.edu/>). Another example is the Gramene database (<http://www.gramene.org>, Ware *et al.* 2002; Jaiswal *et al.* 2006a,b), which holds over 7300 entries for rice, maize, barley, oat and wild rice.

1.2 Usefulness and Verifiability

QTL detection results are useful in so much as they can be applied in plant and animal breeding programs and in positional cloning and characterization of genes. All of these applications are relatively costly and are severely hampered when false QTL detections are pursued. False positive error-rates must be kept low so that the apparent successes of QTL mapping may be translated into true successes in the areas of marker assisted selection and genetic characterization. It is not possible to eliminate all false detections because QTL analysis relies on statistical methods, but a desirable QTL detection strategy should at least keep false positives down to the nominal value of the chosen significance level.

The traditional approach to find the physical location of a gene controlling a Mendelian trait is to begin with a known gene product (a protein with a known function), then determine the protein's amino acid sequence and use it to isolate the gene. This approach is not practical for mapping complex traits because there is usually no information about what proteins could be involved. The aim of positional cloning is to construct a molecular map by using a genetic map as the starting point. Therefore QTL mapping is performed first. Then overlapping segments of DNA are copied from a region which is defined by a confidence interval for the QTL location. Genetic procedures such as mutational analysis are applied to authenticate the selected gene locations. For detailed discussions of positional cloning see Arondel *et al.* 1992; Tanksley *et al.* 1995; Vladutu *et al.* 1999; Jander *et al.* 2002; Morgante and Salamini 2003.

When two or more QTL exist on the same chromosome, methods which assume a single QTL can reveal a false or 'ghost' QTL whose map location is different from any of the true QTL locations (Knott and Haley, 1992). This lack of accuracy hinders positional cloning. Several authors have proposed multiple-QTL models (for example

Kao *et al.*, 1999). This thesis also proposes a multiple-QTL model which is robust against ghosting. Genetic maps having good precision will aid also positional cloning because the range of DNA segments to copy and test will then be relatively small.

Since ancient times, plant and animal breeders have found that they could improve the quality of their stocks by selecting individuals having the desirable phenotype to be the parents of the next generation. The paper by Soller and Medjugorac (1999) provides a good overview of how genetic data can be used together with phenotype data to enhance breeding programs. The paper by Soller and Medjugorac (1999) also describes how the work of early pioneers like Sewall Wright, Sir Ronald Fisher and Jay Lush contributed to developing a framework for applying QTL analysis in breeding programs. Marker assisted selection (MAS) is meant to fine-tune selective breeding schemes by using both phenotypic and genetic characteristics to select parents.

Marker assisted selection exploits the fact that a trait controlling QTL can be indirectly selected by selecting for genotypes of a marker that is located very close to it. Indirect selection is made possible by the fact that tightly linked genes (genes located very close together on the same chromosome) tend to be transmitted together in generations. The aim is to increase the frequency of the desired QTL alleles from generation to generation. If linked QTL can be detected close to known markers, then breeders have an indication of which markers will be useful for MAS. Marker assisted selection depends on QTL analysis results. However, MAS has the advantage that the exact location of the QTL does not need to be estimated. Despite this advantage, there has been very limited success in MAS breeding programs. For example, Milhaljevic *et al.* (2004) noted that in most published experiments on MAS only about half of the QTL under selection actually contributed to the realized selection response.

The poor performance of MAS and other applications of QTL mapping output have caused researchers to be very cautious when interpreting and using QTL analysis results. Consequently, the strategies of independent validation and cross validation

(see Visscher *et al.*, 2000; Bohn *et al.*, 2001) and Meta Analyses (see Goffinet and Gerber, 2000; Xu, 2003), have been used to assess uncertainty in QTL mapping. Still, there are many cases where researchers conducting independent experiments have found agreement on the existence and locations of certain QTL. Therefore QTL analysis remains a popular research area because it can allow the detection of genes having large effects and because it has the potential to detect QTL of moderate to small effects, provided that strategies for reducing the uncertainties which plague the data analysis are found. Developing robust QTL analysis techniques is also a worthy endeavour because it provides a means to more effectively use the vast amount of genetic marker data that is being made available through recent genetic mapping projects.

1.3 The Challenges of QTL Mapping

Some of the challenges that affect model development in the context of QTL analysis are described below.

1. There is uncertainty about how QTL genotypes contribute to trait expression. Consequently, there is uncertainty about the conditional distribution of the trait given a particular QTL genotype. The most common approach is to assume that a trait is Normally distributed within samples of individuals who have the same genotype at the selected loci and who come from similar environments.
2. The QTL locations are unknown and QTL genotypes cannot be observed. Therefore the trait distribution conditional on a QTL genotype must be found by applying the theorem of conditional probability with assumptions about the probability of each QTL genotype given each marker genotype.

3. In order to detect association between marker and QTL, the chosen experimental design must capture information about the probability of each QTL genotype given each marker genotype. In order to map QTL, it must capture information for linkage. To detect recombination between two loci, the parent under consideration must be heterozygous at both loci. The QTL genotypes in all parents are unknown, therefore a suitable experimental design must allow inferences to be made about the parental QTL genotypes given their observed marker genotypes. This is necessary to allow assessments to be made about the probability that a particular offspring is the result of recombinations between parental marker and QTL loci. Parents from crosses of inbred lines divergent in trait values as well as in their marker genotypes are often used with plant and some animal species. For species in which inbreeding is not feasible, family studies must be used in order to detect recombination. Still, there is uncertainty about the probabilities of different QTL genotypes within the marker-classes generated by any chosen sampling design.
4. Most complex traits are conditioned by more than one locus and there is uncertainty about the number of loci involved. The most common approach to this problem is to assume a fixed number of loci. However, models which assume a single QTL often suffer from ghosting (false detections), while models which assume multiple QTL often suffer from identifiability problems. Otto and Jones (2000) discuss some of the limitations of techniques that attempt to estimate the true number of QTL controlling a trait. Where multiple QTL exist, there may also be a need to separate the effects of different QTL.
5. Traditionally, the main emphasis has been on estimating non-interaction terms in a linear model for QTL effects. Specific contrasts of conditional trait means, called additive and dominance genetic effects, receive much attention in the

literature because they have convenient interpretations (Falconer and Mackay (1996)). However, in fitting a linear model, the precise choice of contrasts is not particularly important except for removing the singularity of the model matrix. Any suitable contrasts may be used, and after fitting, any other desired contrasts may then be obtained from the fitted means provided that the number of simultaneous contrasts is not greater than the rank of the model matrix. It is also noteworthy that common breeding designs produce rank-deficient systems. For example, Backcross designs produce rank-deficient systems that do not allow additive and dominance effects to be separated.

6. There is a possibility that interactions may exist between loci. The number of interactions is unknown.
7. QTL expression can also be influenced by non-genetic factors. There are often problems distinguishing genetic effects from environmental effects and evaluating interactions between genetic and non-genetic factors.
8. The heritability of a trait will also affect the power of QTL detection. Heritability is a measure of how much of the total trait variation is due to a genetic component. Genes controlling traits with low heritability may be difficult to detect via marker-trait association because most of the variability seen will tend to be absorbed into the random error.
9. Often, estimated QTL effects are confounded with functions of QTL genotype probabilities (which are functions of recombination fractions). This confounding creates bias in the estimated QTL effects. If a QTL is located extremely close to a marker the magnitude of the estimated effect will be biased downwards and so the QTL will be difficult to detect. If a QTL coincides with a marker, then it may go undetected when models assume that markers have no effect on

trait value (the neutral marker assumption).

10. A dense map of markers can improve the accuracy and precision of QTL mapping but there is a point where adding more tightly linked markers does not add any more information. Fitting a regression based upon very dense marker-map requires large sample sizes to compensate for the degrees of freedom needed to estimate the large number of parameters generated. Depending on the species and the trait being studied, obtaining very large sample sizes may not be possible. The fact that genotypes at linked markers do not segregate independently may reduce the utility of overly dense marker-maps. The explanatory variables may be highly correlated when genotypes of tightly linked markers are used in regression models. If the map is too dense the resulting model matrix is likely to be ill-conditioned, leading to poor parameter estimates and more false QTL detections. If background markers are too close to the position being tested then they can absorb the QTL effects due the high correlation between marker and QTL genotypes. Davarsi and Soller (1994) modelled the cost of raising individuals and scoring markers (for use in a marker-QTL experiment) as a function of marker spacing and the number of scored individuals in order to access how these factors affected the ability to detect QTL. They found that a marker spacing between 20 to 30 centiMorgans (cM) generally tends to be optimal and that any marker spacing below 10 cM is generally not cost effective.
11. The quality of QTL mapping results is affected by sample size. If the sample size is too small some genotypes may not be observed or the counts in some genotype classes may be too small to provide reliable estimates of recombination fractions. There are two sample size problems in QTL mapping. The first problem is that biological, ethical and budgetary constraints can make it difficult to obtain large sample sizes. This problem becomes compounded when the trait

of interest is rare. Several factors can affect whether a particular sample size is adequate. These factors include, but are not limited to, the breeding design, marker density, the number and location of QTL, the size of QTL effects and the heritability of the trait, and the data analysis and estimation techniques used. The second problem is that there is currently no established procedure for combining these factors to form criteria for calculating what sample size is large enough to yield reliable QTL mapping results (Frankel, 1995; Belknap, 1998).

1.4 Review of the model-development literature

Tests for differences between conditional means, analysis of variance, linear regression, generalized linear regression, mixed models, likelihood methods, empirical methods, nonparametric methods and Bayesian methods have all been used to analyse quantitative trait loci. Many of these methods have been in existence for decades but the availability of high speed computers has opened up new ways of using them. The most widely used models are those based on extensions of the Fisher (1918) linear model.

Single marker methods test for association between the trait and the genotypes at each marker, independently, not considering genotypes at any other marker. Single marker analysis may be based on t-tests for differences between means, simple regression or one-way analysis of variance (Soller *et al.*, 1976; Stuber *et al.*, 1987; Edwards, 1987) and likelihood ratio tests (Weller, 1986).

Kearse and Hyne (1994) suggested a generalized least-squares regression approach which uses all markers on a chromosome to improve the precision of single-marker analysis. The differences in mean trait value of the genotypes at each marker locus form the vector of response variables. The vector of explanatory variables comprises

the distance between each marker and a putative QTL. The analysis is repeated for a series of QTL locations along a chromosome. This procedure is also called ‘multipoint mapping’. Critical values are based on the assumption of a chi-square distribution for the residual sum of squares.

Thoday (1961) was among the first to use a pair of adjacent markers to estimate QTL effects and position. The process of detecting a QTL by simultaneously conditioning on a pair of markers lying on either side of it later became known as Interval Mapping. Lander and Botstein (1989) proposed a likelihood-based approach to interval mapping which assumed an underlying Normal trait distribution for individuals having the same QTL genotype. Like Thoday, Lander and Botstein modelled QTL effects by conditioning on the genotypes at a pair of adjacent markers. However, the latter used maximum likelihood estimation via the EM algorithm (see Dempster *et al.*, 1977) to estimate QTL effects.

In an attempt to reduce the computational burden of maximum likelihood estimation for interval mapping, Bridges and Knapp (1990), Haley and Knott (1992) and Martinez and Curnow (1992) advocated the use of regression methods. They proposed carrying out regressions at several putative QTL locations and taking regression estimators at the location that maximizes the regression correlation coefficient as approximations to the desired maximum likelihood estimators.

Whittaker *et al.* (1996) used contrasts of trait means within marker groupings to show that the location and effect of an isolated QTL (having no additional QTL in adjacent intervals) can be estimated from a regression of phenotype on marker type, without the need for numerical search procedures.

The interval mapping approach of Lander and Botstein (1989) and its regression approximations all assume that either there is a single QTL between the markers under consideration, or that there is no QTL anywhere. This leads to a single Normal distribution under the null hypothesis and a Normal Mixture under the alternative

hypothesis. The likelihood ratio test in this situation amounts to a test of departure from normality of the trait distribution. If a single QTL exists within the specified interval, then additional linked QTL will increase the number of mixing components and will contribute to the sampling variance. Additional QTL can also lead to pooling of effects causing biased estimates. If there is no QTL within the specified interval, the presence of linked QTL outside the testing interval will lead to a null distribution which is a normal mixture rather than a single normal distribution. This could lead to false detections if departure from normality of the trait distribution is taken, on its own, to indicate the presence of a QTL. The removal of outliers may be undesirable if the true distribution is a mixture because outliers may result from rare combinations of genotypes which are in fact valid for the mixture. Similarly, transformations to normality may not be desirable if the underlying distribution is in fact a normal mixture. Such transformations could hamper the detection of an existing QTL.

Realizing that the likelihood ratio test (LRT) for normality of the trait distribution does not necessarily constitute a test for a QTL within a specified interval, researchers needed methods for assessing the results of interval mapping in terms of whether a QTL was detected.

Bootstrap and permutation methods are useful where an estimator of a statistical property of interest is available but its distribution is unknown. Churchill and Doerge (1994) proposed an empirical method for calculating approximate significance thresholds (critical values) against which to compare test statistics for QTL mapping. They proposed that critical values should be derived from Monte Carlo tests based on the empirical distribution of these statistics. The idea was to draw samples that would be representative of the null hypothesis of no QTL within a selected marker interval. Churchill and Doerge (1994) showed that such a sample can be obtained by randomly assigning an observed trait value, without replacement, to each sampled individual

while leaving its genotype unchanged. Although Churchill and Doerge (1994) mention the possibility of extending the permutation tests to the problem of detecting multiple QTL, this was not explored. In a similar vein, Visscher *et al.* (1996b) suggested using the bootstrap methodology of Efron (1979), with critical values based on the empirical bootstrap distribution of the test statistic. Despite their heavy computational requirements, these resampling methods are widely used because they are simple to implement.

Other researchers used semi-parametric and non-parametric methods to allow for the fact that a null distribution could be non-normal. For example, Kruglyak and Lander (1995) proposed some Wilcoxon rank-based tests of genetic effects. Zou (2001) considered a single-QTL framework and applied the Kruglyak and Lander (1995) rank based test to estimate quantitative trait effects. However her simulation results showed that for both normal and non-normal data, the non-parametric test performed similarly to the Normal regressions of Haley and Knott (1992). Zou (2001) also proposed a semi-parametric approach to interval mapping, based upon the exponential tilt model of Anderson (1979). The exponential tilt model was found to be susceptible to identifiability problems similar to those that plague parameter estimation in normal mixtures.

A simple procedure that has proved to be very robust against false detection (ghosting) involves carrying out standard interval mapping with flanking markers to absorb the variance of background QTL. This procedure, called Composite Interval Mapping (CIM), was independently proposed by Rodolphe and Lefort (1993), Zeng (1993, 1994), and Jansen and Stam (1994). These authors showed that CIM can aid the separation of pooled QTL effects provided that Haldane's map function holds reasonably well, and provided that no extra QTL lie within the intervals bracketed by the nearest flanking markers. However, if Haldane's map function holds and extra QTL exist within intervals adjacent to the testing interval, then CIM cannot separate

the effects of QTL from the resulting region stretching over three adjacent intervals. Therefore, false detection rates of CIM (traditional LRT based on χ_1^2 distribution) are well controlled only when the testing interval is isolated.

Zeng (1993, 1994) proposed models that allow for multiple QTLs and Hayes *et al.* (1993); Jiang and Zeng (1995) considered the problem of QTL-environment interactions.

Obtaining standard errors for estimates of model parameters in CIM and other mixture models has always been a challenge. Bootstrapping has been proposed as a method of addressing this problem in QTL mapping problems (Visscher *et al.* (1996b)). In order to obtain asymptotic standard errors of model parameters in QTL mapping, Kao and Zeng (1997) proposed formulae for calculating the conditional observed information matrix. Kao and Zeng (1997) did not provide formulae for calculating the Fisher information because they did not take the expectation of the conditional information matrix. Also, they did not explore the idea of statistical tests for QTL based upon the asymptotic distribution of the maximum likelihood estimators. Instead, they used a LOD score of 1.5 (as suggested by Lander and Botstein, 1989) to determine the threshold value for rejection of the null hypothesis. The element-wise approach used by Kao and Zeng (1997) is not sufficiently general to make evaluation of the information matrix both practical and accessible for any Multinomial mixture of Normals.

Making the information matrix practical to calculate for mixture likelihoods is a problem that has received much attention in the statistical literature. Hill (1963) used a power series expansion to simplify the Fisher information matrix for a mixture of two univariate Normal distributions having equal variances. Behboodian (1972a, 1973) provided numerical methods for evaluating the Fisher information matrix for mixtures of two Normal distributions and for mixtures of two Exponential distributions.

For Multinomial mixtures of Normals, McLachlan and Basford (1987, page 47)

approximated the observed information matrix, in terms of the gradient vector of the log-likelihood function. In a simulation study, they found that, when compared with standard errors obtained by bootstrapping, this approximation tended to overestimate the variance of the parameter estimates. Therefore Basford *et al.* (1997) recommended using standard errors based on bootstrap methods rather than using standard errors based on the observed information formulae of McLachlan and Basford (1987).

The development of Markov Chain Monte Carlo (MCMC) methods has facilitated Bayesian estimation for mixture models (Casella and George 1992; Smith and Roberts 1993; Diebolt and Robert 1994; Carlin and Lewis, 1996, pages 60, and 159-197; Richardson and Green 1997). Subsequently, various researchers have applied Bayesian approaches to parameter estimation in QTL mapping (Guo and Thompson, 1992; Satagopan *et al.*, 1996; Satagopan and Yandell, 1996; Hoeschele *et al.*, 1997; Ball, 2001).

The Bayesian approach is appealing because it allows the number of mixing components (the number of QTL) to be explicitly included as an unknown parameter in the model, and it also allows estimation of marginal posterior probabilities for the parameters. However, non-identifiability problems can arise with the Bayesian approach to finite mixture modelling and the MCMC method can also suffer from convergence issues (Diebolt and Robert, 1994; Robert, 1996).

This thesis does not explore Bayesian methods for QTL mapping, instead it focuses on the problem of improving hypothesis testing for parameters in mixture models under the maximum likelihood framework. The next section outlines the main contribution of this thesis.

1.5 Contribution of this Thesis

This thesis explores and develops mathematical and statistical techniques that are tailored towards extracting a desired type of information from samples of genetic (DNA) data coupled with measurements of a specific trait. The desired information is any that will enable detection of genes associated with the trait, estimation of their genetic effects and, in the presence of linkage, estimation of their genetic location.

The existing strategies for inferring QTL from multiple regressions of trait value on marker genotypes are consolidated and formalized. Improved hypothesis tests for Composite Interval Mapping are proposed.

A new extension to Composite Interval Mapping is developed. The proposed model, named Robust Interval Mapping Version One (RIM1), may be viewed as a more robust extension of CIM. The RIM1 model fits exactly three putative quantitative trait loci (QTL) and it uses maximum likelihood estimation to obtain estimates of model parameters. Applications to simulated and real data show that these methods have strong power to detect QTL while dramatically decreasing the rate of false detections.

New, very flexible, matrix formulae are developed, allowing exact and convenient calculation of both the Observed and Fisher information matrices in the context of Multinomial mixtures of Univariate Normal distributions. Standard errors based on these formulae are then used to create tests which reduce false detections in CIM while retaining power to detect QTL.

1.6 Thesis Layout

A brief overview of the literature was presented in this introductory Chapter. In Chapter 2, there is an overview of classical quantitative genetics definitions of genetic effects, linkage and sampling designs. Chapter 3 looks at the mixture structure of

line-cross designs and highlights aspects of that structure which could carry information for model development and hypothesis testing. Chapter 4 reviews the Normal regression approach to QTL mapping.

Chapter 5 is a long Chapter where new techniques are introduced: information matrix formulae are introduced in Chapter 5 as well as an extension to composite interval mapping, named Robust Interval Mapping Version One (RIM1). Chapter 6 tackles the derivation of the information matrix formulae which were presented, without proof, in Chapter 5. Although the detailed mathematical proofs given in Chapter 6 are rather tedious, the proofs are necessary because they show why the proposed formulae constitute an exact evaluation of the information matrix.

In Chapter 7, the proposed methods are applied to simulated data. Extensions and applications of the proposed methods to some real data are given in Chapter 8. The final chapter (Chapter 9) summarizes the results of this thesis and discusses areas for further research.

Chapter 2

Linkage, Breeding Designs and Genetic Effects

This chapter presents an overview of classical Quantitative Genetics Definitions. The first section focuses on linkage, recombination probabilities, mapping populations and experimental designs. In the later sections we look at the definitions of *genetic effects* as well as useful properties resulting from these definitions.

2.1 Linkage and Recombination Fractions

Two genes are said to be *linked* if they are located on the same chromosome. The proximity of linked genes to each other affects their probability of being transmitted together from parent to offspring. In meiosis (sperm or egg production) homologous (similar) chromosomes may overlap and exchange genetic material. This process is called *recombination* or *crossing-over*. Crossover is more likely to occur in the interval between linked genes that are located far apart than between closely linked genes.

The *recombination fraction* between two loci is the probability that there will be an odd number of crossovers between them. Even numbers of crossover are generally

not considered because they cannot be observed.

Consider a set of chromosomes for which a number of markers have been mapped, and which contain an unknown number of QTL at unknown locations. For inference about the properties of these QTL, we need to determine the probability of each (multi-locus) QTL genotype conditioned on each marker genotype. Assessment of the recombination fraction between pairs of loci enables us to write down expressions for the probability of multi-locus genotypes and expressions for the probability that a QTL allele is transmitted given that certain marker alleles are transmitted.

Consider three linked loci in the order A - B - C and let r_{AB} and r_{BC} be the probabilities of recombinations between loci A and B and loci B and C respectively. Let r_{AC} be the recombination fraction between loci A and C . Recombinations in the different intervals may not occur independently (see Ott 1991). For instance, when the loci are closely linked, a recombination in one interval may reduce the likelihood of recombination in an adjacent interval.

In genetics, lack of independence between crossover events in different intervals is called *crossover interference* or *recombinational interference*. Under independence, a double recombination occurs with probability $r_{AB}r_{BC}$. If its true probability is π_{11} then the *coefficient of coincidence* is defined as

$$c = \frac{\pi_{11}}{r_{AB}r_{BC}}$$

and recombinational interference is measured by $1 - c$. In the case of complete interference $c = 0$. When $c = 1$ there is no interference. Positive interference results when $c < 1$ and there is negative interference when $c > 1$.

Define

$$\pi_{11} = P(\text{recombination in both intervals}) = c r_{AB} r_{BC} \quad (2.1)$$

$$\pi_{10} = P(\text{recombination in interval } A - B \text{ only}) = r_{AB}(1 - c r_{BC}) \quad (2.2)$$

$$\pi_{01} = P(\text{recombination in interval } B - C \text{ only}) = r_{BC}(1 - c r_{AB}) \quad (2.3)$$

$$\pi_{00} = P(\text{no recombination in either interval}) = 1 - r_{AB} - r_{BC} + c r_{AB} r_{BC} \quad (2.4)$$

Recombinations between A and C can occur in two ways. Either there is recombination in the interval $A - B$ and no recombination in the interval $B - C$ or there is no recombination in the interval $A - B$ and recombination in the interval $B - C$. This leads to the general three-locus addition formula for recombination fractions given in Equation (2.5).

$$r_{AC} = \pi_{10} + \pi_{01} = r_{AB} + r_{BC} - 2c r_{AB} r_{BC} \quad (2.5)$$

The expected number of recombination events between two loci is called the *genetic distance* between them and is measured in Morgans. Genetic map functions are used to translate recombination fractions into genetic distances.

The Haldane (1919) map function is the most commonly used genetic mapping function. It assumes that recombination events occur independently of each other (no interference) and that they occur as points of a Poisson process along each chromosome. Under Haldane's assumptions, the number of crossovers between two loci x Morgans apart has a Poisson(x) distribution. Therefore, Haldane's map function to convert the recombination fraction r_{AB} to a genetic distance is

$$x = -\frac{1}{2} \log(1 - 2r_{AB}).$$

Real data does not usually support the idea of constant levels of interference in all intervals along a chromosome. However, for simplicity, most common map functions assume a fixed value for c in the addition formula for recombination fractions. For example, $c = 1$ for Haldane's addition formula. By setting $c = 2r_{AB}$ Kosambi (1944)

produced an addition formula that allows for non-constant interference. Detailed descriptions of these and other map functions may be found in Quantitative Genetics texts (see, for example, Ott 1991, pages 14-19 and 120-129; Liu 1997, pages 318-329). For recombination probabilities up to 0.1, most map functions give similar estimates of the map distance (see, for example, Table 10.9 on page 329 of Liu 1997). For, example, when the recombination fraction is less than or equal to 0.1, the Morgan, Haldane, Kosambi, Felsenstein, Carter-Falconer map functions yield approximately the same map distances. Therefore, for very dense maps, the Haldane assumption does not cause too much concern. It is more of a concern when map density is low.

2.2 Breeding designs

In order to detect association between marker and QTL, the chosen breeding design must capture information for linkage. The most common breeding designs allow assessments to be made about recombination probabilities, genotype probabilities and about the probability of putative QTL genotypes given any marker genotype. This section gives a brief overview of some commonly used experimental populations and breeding designs. Here we are considering diploid organisms only.

In the following discussion, the founding parents (first parents) from which inbred designs are created are denoted by P1 and P2 respectively. The P1 and P2 lines are assumed to be homozygous at all loci. Additionally, the alleles at any locus in the P1 line are assumed to be different from the alleles at the same locus in the P2 line. For convenience, we refer to an allele from P1 line as a ‘high’ allele, and we refer to the corresponding allele from P2 line as a ‘low’ allele. We denote high and low alleles, respectively, by uppercase and lowercase Roman letters. We refer to a (single-locus) genotype from the P1 line as a ‘homozygous-high’ genotype and we refer to the corresponding P2 genotype as a ‘homozygous-low’ genotype.

Inbreeding without selection

1. Backcross: B1 or B2

Two diverging, inbred lines (P1, P2) are crossed and the resulting offspring (F1) are back-crossed with the first parental line (P1) to form the B1 backcross or with the second parental line (P2) to form the B2 line (see Figure 2.1(a)). All parents (F1 or P1) or (F1 or P2) are completely informative for linkage. At any single locus, only two distinct genotypes are possible and they occur with equal probability. At any locus only the homozygous-high and the heterozygous genotypes are possible in the B1 backcross. Likewise, at any locus, only the homozygous-low and the heterozygous genotypes are possible in the B2 backcross. Consequently the genotype probabilities in these backcross populations do not occur in Hardy-Weinberg proportions (see, for example, Hartl and Clark, 1997). Nevertheless, the backcross design has the advantage that the genotype phase (that is, the sister-chromatid locations of alleles in a multi-locus genotype) of all backcross individuals can be determined.

2. **Second filial line: F2 intercross.** Two diverging, inbred lines are crossed to form the F1 line. Then the F1 is ‘selfed’ or made to undergo brother-sister mating to produce the F2 line (see Figure 2.1(b)). This breeding design is also referred to as an F2 *intercross* or simply an *intercross*. One advantage of the F2 design is that its genotypes occur in Hardy-Weinberg proportions. However, only the homozygous F2 individuals are informative for linkage (they allow the origin of the parental alleles to be determined so allowing recombinantions to be identified without ambiguity). Consequently, the homozygous F2 individuals they are the only F2 individuals whose genotype phase can be determined.

3. **Second backcross line (BC2).** A second backcross line is formed by crossing the F2 line with the first parental line or with the second parental line. Figure

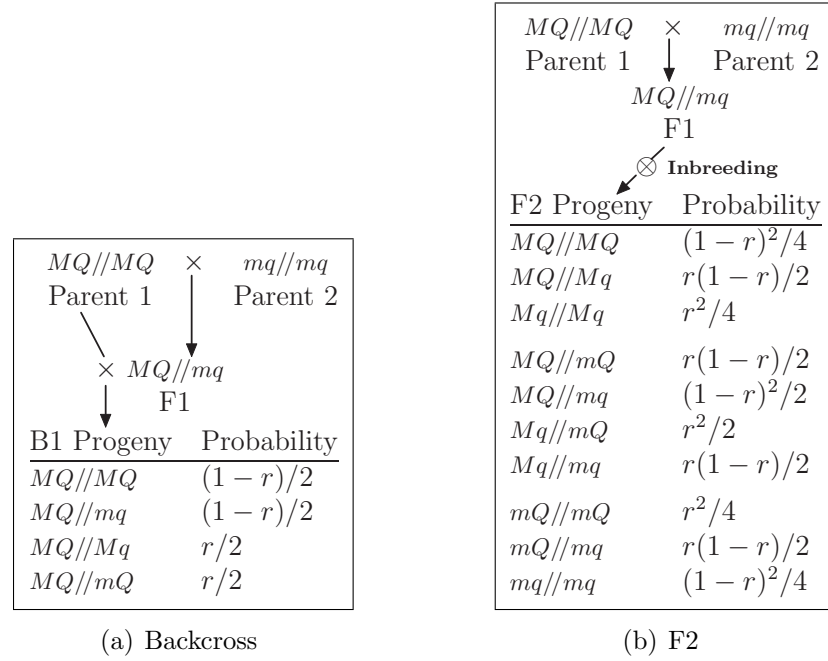


Figure 2.1: Definitions of backcross (from parent one) and F2 progeny for a single marker locus (M) and a QTL locus (Q) that are r recombination units apart. In the F2 population, there are nine distinct two-locus genotypes – in the F2, the genotype $MmQq$ has two possible phases: $MQ//mq$ and $Mq//mQ$.

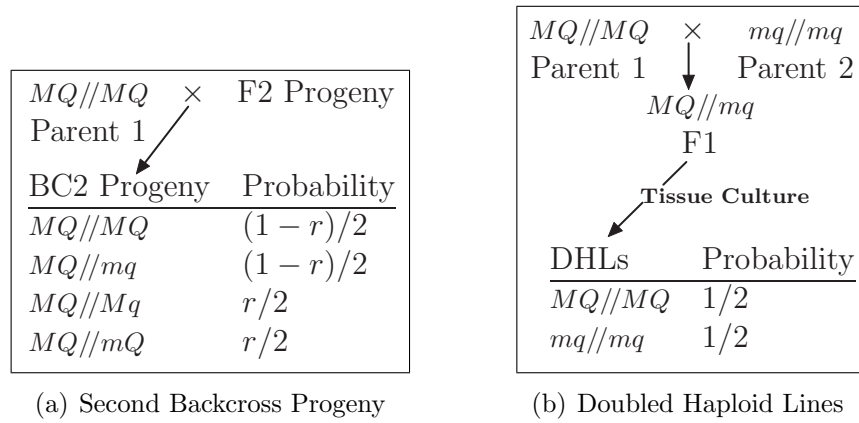


Figure 2.2: Definitions of a Second Backcross and Doubled Haploid Lines for a single marker locus (M) and a QTL locus (Q) that are r recombination units apart.

2.2(a) illustrates the case where the backcross is made with the first parental line.

4. **Doubled haploid lines (DHLs).** Doubled haploid lines are formed by chemically treating some organisms to cause them to replicate producing identical copies of themselves (see Figure 2.2(b)). This technique is only practical in a few species, for example, Zebrafish and *Drosophila*.
5. **Advanced intercross lines: AIL or $F(t)$.** Advanced intercrossed lines are formed by repeated selfing or brother-sister mating of F1 over $t - 1$ generations. These are created by random mating of F1 individuals followed by random mating in all subsequent lines over $t - 1$ generations to produce F2. Davarsi and Soller (1995) showed that advanced intercross lines generate more recombination events than F2 or Backcross designs. By assuming that crossovers occur independently in adjacent intervals, Davarsi and Soller (1995) derived the following formula for the recombination fraction in the $F(t)$ in terms of a recombination fraction (r) in the F2 population:

$$r_t = \frac{1 - (1 - r)^{t-2}(1 - 2r)}{2}.$$

6. **Repeatedly backcrossed line.** If the offspring from a backcross are repeatedly mated with the original parents for a specified number of generations, then the resulting cross is called a repeatedly backcrossed line.

Inbreeding with selection

1. **Recombinant inbred lines (RIL).** Recombinant inbred lines are produced by inbreeding with selection of recombinants. The F2 line is taken through several generations of ‘selfing’, with selection of recombinant individuals for breeding at each stage. This design provides a method for replication of these recombinant

individuals when asexual reproduction is not possible. Recombinant inbred lines have essentially no within-line genetic variance, but the variance between lines is considerable because each RIL represents a different multi-locus genotype.

2. **Nearly isogenic lines (NIL).** Nearly isogenic lines are formed by repeated back-crossing with selection followed by at least one generation of ‘selfing’ or sib-mating. A donor parent is crossed with an inbred line to form an F1 line. The F1 line is then backcrossed to the inbred line for several generations. Then the individuals in the final generation are sib-mated or ‘selfed’ to form a nearly isogenic line.

Outbred designs

1. Experiments orchestrated to extract desired information from specific outbred populations are called, collectively, outbred designs. These include sib-pair designs, relative pair designs, family triads and case-control designs.

Certain outbred designs require QTL mapping techniques that are quite different from those used with inbred designs. However, some of the methodology for analysing QTL in outbred designs are extensions of those used with inbred designs (see Lynch and Walsh, 1997, Chapters 16-18). This thesis looks at methodology for detecting QTL in experimental populations, assuming inbred line-cross designs and diploid organisms.

2.3 Genetic effects

2.3.1 Additive, dominance and epistatic effects

In the Quantitative Genetics literature, the value of a trait (or phenotype) is called the *phenotypic value*. Likewise, the part of the phenotypic value that is attributable

to an individual's genotype is called the *genotypic value*. In addition, the expected values of specific contrasts of mean trait-value amongst the genotype classes (within a study population) are called *genotypic effects*.

Each genotypic effect measures the contribution of a particular source of genetic variation to the expected value of a specific trait given a specific genotype. Both the size and the direction of each genetic effect depend on the distribution of the trait within the study population as well as the population the gene and genotype probabilities.

Fisher (1918) defined additive and dominance effects in a linear model for the expected value of a trait given a single-locus genotype. Fisher also partitioned the trait variance according to genetic and environmental sources, with the genetic variation further partitioned into additive and dominance components. Cockerham (1954) and Kempthorne (1954) independently extended Fisher's model to include more than one locus. This section outlines the Cockerham-Kempthorne definitions for additive, dominance and epistatic genetic effects.

Suppose that a single locus M has v distinct alleles and denote them by M_1, \dots, M_v respectively. Assume diploid organisms. Suppose also, that

$P(M_i)$ is the probability of allele M_i in the population;

$P(M_i M_j)$ is the population probability of genotype $M_i M_j$;

$P(M_i | M_i M_j)$ is the probability of the allele M_i among all alleles belonging to genotype $M_i M_j$.

These allele and genotype probabilities have the properties given in Equations (2.6)

to (2.9) below.

$$P(M_i) = P(M_i M_i) + \frac{1}{2} \sum_{j \neq i} P(M_i M_j) \quad (2.6)$$

$$\sum_{i=1}^v P(M_i) = 1 \quad (2.7)$$

$$\sum_{i=1}^v \sum_{j \leq i} P(M_i M_j) = P(M_i M_i) + \sum_{j \neq i} P(M_i M_j) = 1 \quad (2.8)$$

$$P(M_i | M_i M_j) = \begin{cases} \frac{1}{2}, & \text{if } i \neq j \\ 1, & \text{if } i = j \end{cases} \quad (2.9)$$

Note that the conditional probability, $P(M_i | M_i M_j)$, is also the probability that an individual with genotype $M_i M_j$ will transmit allele M_i to an offspring.

Denote the trait value by the random variable y . Also, let $E(y | M_i)$ represent the mean trait-value of individuals having allele M_i at a single locus and let $E(y | M_i M_j)$ represent the mean trait-value of individuals with genotype $M_i M_j$ at that locus. We denote mean trait-value (in the population under study) by μ , where

$$\mu = E(y) = \sum_{i'=1}^v \left(P(M_{i'} M_{i'}) E(y | M_{i'} M_{i'}) + \sum_{j \neq i'} P(M_{i'} M_j) E(y | M_{i'} M_j) \right). \quad (2.10)$$

The *additive effect* of an allele is the difference between the mean trait-value of individuals having that allele and the population mean trait-value. It may be interpreted as the phenotypic value associated with a gene that is passed on to an offspring (see, for example, Falconer and Mackay, 1996, pages 112-117). The additive

effect of allele M_i is defined as

$$\begin{aligned}
\alpha_{M_i} &= E(y | M_i) - \mu \\
&= \sum_{j=1}^v P(M_i M_j | M_i) E(y | M_i M_j) - \mu \\
&= \sum_{j=1}^v \frac{P(M_i | M_i M_j) P(M_i M_j)}{P(M_i)} E(y | M_i M_j) - \mu \\
&= \frac{P(M_i M_i)}{P(M_i)} E(y | M_i M_i) + \frac{1}{2} \sum_{j \neq i} \frac{P(M_i M_j)}{P(M_i)} E(y | M_i M_j) - \mu
\end{aligned} \tag{2.11}$$

This implies that

$$\begin{aligned}
\alpha_{M_i} &= \left(\frac{1}{P(M_i)} - 1 \right) P(M_i M_i) E(y | M_i M_i) \\
&\quad + \left(\frac{1}{2P(M_i)} - 1 \right) \sum_{j \neq i} P(M_i M_j) E(y | M_i M_j) \\
&\quad - \sum_{i' \neq i} \sum_{j \neq i} P(M_i' M_j) E(y | M_i' M_j).
\end{aligned} \tag{2.12}$$

Equation (2.12) is an example of a *contrast*: a linear combination of conditional trait means.

The additive effect can be estimated using the coefficients from a regression of the trait value on the number of copies of target alleles in the genotype. A direct consequence of the definition of additive allelic effect (as given in Equation (2.11)) is that the mean value of the additive allelic effects at a locus is equal to zero:

$$\sum_{j=1}^v P(M_j) \alpha_{M_j} = 0 \implies \alpha_{M_i} = -\frac{1}{P(M_i)} \sum_{j \neq i} P(M_j) \alpha_{M_j}. \tag{2.13}$$

So far, we have discussed the additive effect of a single allele (the additive allelic effect) at a single marker M . Now we turn to the additive effect of a genotype at locus M . Distinct alleles M_i and M_j of the gene at locus M are called codominant alleles if the alleles can be individually identified in the heterozygous genotype $M_i M_j$ ($i \neq j$). The ability to distinguish loci, and the ability to identify alleles at those loci,

depends on the instrumentation and processes used to classify DNA segments (see for example Liu, 1997, pages 62-82).

If the heterozygous genotype M_iM_j is expressed (on the classification instrument) in a manner that is identical to M_iM_i , then M_i is said to display complete dominance over M_j . Likewise, if it is expressed as M_jM_j then, then M_j is said to display complete dominance over M_i . If one allele is completely dominant over the other, then the marker technology in use does not allow the heterozygous genotype to be distinguished from one of the homozygous genotypes.

The *breeding value* or *additive (genotypic) effect* of a genotype is defined as the sum of the additive effects of its component alleles. Therefore, this definition assumes that the different genotypes and their component alleles can be separately identified. Let us assume that distinct alleles M_i and M_j are codominant. Then the additive effect of genotype M_iM_j is equal to

$$a_{M_iM_j} = \alpha_{M_i} + \alpha_{M_j}. \quad (2.14)$$

For homozygous genotypes, the additive effect has the form

$$\begin{aligned} a_{M_iM_i} &= 2\alpha_{M_i} \\ &= -\frac{2}{P(M_i)} \sum_{j \neq i} P(M_j) \alpha_{M_j} \text{ from Equation (2.13) above} \\ &= -\frac{1}{P(M_i)} \sum_{j \neq i} P(M_j) a_{M_jM_j} \end{aligned} \quad (2.15)$$

The mean of the additive effects of all genotypes at the locus M is called the mean breeding value of M .

The *dominance (genotypic) effect* of genotype M_iM_j is defined as

$$\begin{aligned} d_{M_iM_j} &= E(y | M_iM_j) - \mu - a_{M_iM_j} \\ &= E(y | M_iM_j) - \mu - (a_{M_iM_i} + a_{M_jM_j})/2. \end{aligned} \quad (2.16)$$

Note that this dominance (Equation (2.16)) is distinct from the dominance defined on the previous page, in which genotypes are being classified. This dominance effect represents interaction between two alleles at the same locus. It is that part of the difference in mean trait value (between the subpopulation with genotype M_iM_j and the overall population) which cannot be accounted for by additive effects. Like the mean of the additive effects, the mean of the dominance effects is equal to zero when averaged over a population with genotype probabilities $P(M_iM_j)$.

Genotypic effects associated with interactions between genes at different loci are called *epistatic effects*. The second order epistatic effects (*additive* \times *additive*, *additive* \times *dominance* and *dominance* \times *dominance* effects) are interactions involving two distinct loci. For the definitions of these interaction effects, consider two different loci, M and N . Let M_i and M_j be the i^{th} and j^{th} alleles, respectively, at locus M . Similarly, let N_k and N_ℓ be the k^{th} and ℓ^{th} alleles, respectively, at locus N .

There are four additive \times additive interactions for any pair of loci. Each additive \times additive effect measures the interaction of an allele at one locus with an allele at another locus. The *additive* \times *additive* effect between allele M_i and allele N_k is defined as

$$(\alpha\alpha)_{M_iN_k} = E(y | M_iN_k) - \mu - \alpha_{M_i} - \alpha_{N_k}. \quad (2.17)$$

To calculate $E(y | M_iN_k)$, the following formulas are useful.

$$E(y | M_iN_k) = \sum_{j,\ell} \frac{P(M_iN_k | M_iM_jN_kN_\ell) P(M_iM_jN_kN_\ell)}{P(M_iN_k)} E(y | M_iM_jN_kN_\ell)$$

$$P(M_iN_k) = \sum_{j,\ell} P(M_iN_k | M_iM_jN_kN_\ell) P(M_iM_jN_kN_\ell).$$

The transmission probabilities $P(M_i N_k | M_i M_j N_k N_\ell)$ are given by

$$P(M_i N_k | M_i M_j N_k N_\ell) = \begin{cases} 1 & \text{if } i = j \text{ and } k = \ell \\ 1/2 & \text{if } i = j \text{ and } k \neq \ell \\ 1/2 & \text{if } i \neq j \text{ and } k = \ell \\ 1/4 & \text{if } i \neq j \text{ and } k \neq \ell. \end{cases}$$

There are four *additive* \times *dominance* interactions for any pair of loci. Each *additive* \times *dominance* effect measures the interaction between an allele at one locus and a genotype at the another locus. It is defined as

$$(\alpha d)_{M_i N_k N_\ell} = E(y | M_i N_k N_\ell) - \mu - \alpha_{M_i} - a_{N_k N_\ell} - d_{N_k N_\ell} - (\alpha \alpha)_{M_i N_k} - (\alpha \alpha)_{M_i N_\ell}. \quad (2.18)$$

We may calculate $E(y | M_i N_k N_\ell)$ using the following formulas:

$$E(y | M_i N_k N_\ell) = \sum_j \frac{P(M_i N_k N_\ell | M_i M_j N_k N_\ell) P(M_i M_j N_k N_\ell)}{P(M_i N_k N_\ell)} E(y | M_i M_j N_k N_\ell)$$

$$P(M_i N_k N_\ell) = P(M_i M_i N_k N_\ell) + \frac{1}{2} \sum_{j \neq i} P(M_i M_j N_k N_\ell)$$

$$P(M_i N_k N_\ell | M_i M_j N_k N_\ell) = \begin{cases} 1 & \text{if } i = j \\ 1/2 & \text{if } i \neq j. \end{cases}$$

There is one *dominance* \times *dominance* interaction for any pair of loci. The *dominance* \times *dominance* effect, $(dd)_{M_i M_j N_k N_\ell}$, measures the interaction between a genotype at one locus and a genotype at another locus.

$$\begin{aligned} (dd)_{M_i M_j N_k N_\ell} &= E(y | M_i M_j N_k N_\ell) - \mu - a_{M_i M_j} - a_{N_k N_\ell} - d_{M_i M_j} - d_{N_k N_\ell} \\ &\quad - (\alpha \alpha)_{M_i N_k} - (\alpha \alpha)_{M_i N_\ell} - (\alpha \alpha)_{M_j N_k} - (\alpha \alpha)_{M_j N_\ell} \\ &\quad - (\alpha d)_{M_i N_k N_\ell} - (\alpha d)_{M_j N_k N_\ell} - (\alpha d)_{M_i M_j N_k} - (\alpha d)_{M_i M_j N_\ell} \end{aligned} \quad (2.19)$$

Higher order epistatic effects (those involving more than two loci) may be defined similarly.

To write down the linear model of Cockerham (1954) and Kempthorne (1954), suppose that x is a multi-locus genotype. Then, let M and N index loci in x and let M_i and M_j be alleles at locus M in x . Similarly, let N_k and N_ℓ be alleles at locus N in x .

Let A_x , D_x , $(AD)_x$, $(AA)_x$, and $(DD)_x$ be as defined in Equations (2.20) to (2.24).

$$A_x = \sum_M \sum_i \sum_{j \geq i} a_{M_i M_j}, \quad (2.20)$$

$$D_x = \sum_M \sum_i \sum_{j \geq i} d_{M_i M_j}, \quad (2.21)$$

$$(AA)_x = \sum_M \sum_{N \neq M} \sum_i \sum_k (\alpha\alpha)_{M_i N_k} \quad (2.22)$$

$$(AD)_x = \sum_M \sum_{N \neq M} \sum_i \sum_k \left(\sum_{\ell \geq k} (\alpha d)_{M_i N_k N_\ell} + \sum_{j \geq i} (\alpha d)_{N_k M_i M_j} \right), \quad (2.23)$$

$$(DD)_x = \sum_M \sum_{N \neq M} \sum_i \sum_k \sum_{j \geq i} \sum_{\ell \geq k} (dd)_{M_i M_j N_k N_\ell} \quad (2.24)$$

The term A_x is the sum of the additive effects for each locus in x , while D_x is the sum of all the dominance effects. Likewise $(AA)_x$, $(AD)_x$ and $(DD)_x$ are the sums of the respective second-order epistatic effects. If epistatic effects of order three and higher are negligible, then the model for the conditional trait mean is

$$\begin{aligned} E(y|x) &\approx \mu + A_x + D_x + (AD)_x + (AA)_x + (DD)_x \\ &= \mu + G_x \end{aligned} \quad (2.25)$$

where the genotypic value, G_x , given by

$$G_x = A_x + D_x + (AD)_x + (AA)_x + (DD)_x \quad (2.26)$$

is the part of the conditional trait mean which is due to genetic effects.

2.3.2 Harmonized definitions of genetic effects

The genetic effects of Cockerham (1954) and Kempthorne (1954) are based on orthogonal contrasts. However, they do not represent harmonized definitions of genetic effects because the contrast coefficients (see, for example, Equation (2.12)) are dependent on gene and genotype probabilities, and these probabilities will vary for different populations of a species. Without harmonized definitions for each source of genetic variation, it would not be possible to make valid comparisons between the genetic effects estimated from different studies. Harmonized effects provide a standard for comparison because any specific genetic effect for a study population may be re-expressed as a function of one or more of the (fixed or unchanging) harmonized genetic effects.

The traditional approach for obtaining a fixed basis for comparisons of genotypic effects is to take, as the harmonized definitions, Cockerham-Kempthorne genotypic effects based on an idealized reference population. The chosen reference population is idealized in the sense that it is required to be in both Hardy-Weinberg equilibrium and gametic phase equilibrium and its allelic probabilities are required to be known exactly.

If genotypes at a locus M occur in Hardy-Weinberg proportions, then

$$P(M_i M_i) = P(M_i)^2 \text{ and } P(M_i M_j) = 2P(M_i)P(M_j) \text{ for } i \neq j.$$

Simultaneously requiring the idealized population to have known allelic probabilities and to be in Hardy-Weinberg equilibrium, fixes its single-locus genotype probabilities.

If alleles at two distinct loci M and N are in gametic phase equilibrium (linkage equilibrium), then the probability of the haplotype $M_i N_k$ is given by

$$P(M_i N_k) = P(M_i)P(N_k).$$

Simultaneously requiring the idealized population to have known allelic probabilities and to be in gametic phase equilibrium, fixes its multi-locus genotype probabilities.

The F2 population meets these requirements, and so it is often used as the reference population (see Zeng *et al.*, 2005). The classical approach is to re-express the genotypic effects of the study population in terms of the genotypic effects of a hypothetical F2 population (derived from the same founding parents as the inbred-line being studied). Subsequently, the study population is used to try to obtain estimates for these F2 genotypic effects.

Consider a single locus M with alleles M_1 and M_2 . Let the study population be the B1 backcross and let the reference population be the F2 intercross. Using Equations (2.12), (2.15) and (2.16), we obtain the following expressions for the additive effect (a) and dominance effect (d) of genotype $M_1 M_1$ in the F2.

$$a = \frac{1}{2}(E(y|M_1 M_1) - E(y|M_2 M_2)) \quad (2.27)$$

$$d = \frac{1}{4}(E(y|M_1 M_1) - 2E(y|M_1 M_2) + E(y|M_2 M_2)) \quad (2.28)$$

Likewise, we obtain the following expressions additive effect (a_1) and dominance effect (d_1) of genotype $M_1 M_1$ in the B1 backcross.

$$a_1 = \frac{1}{3}(E(y|M_1 M_1) - E(y|M_1 M_2)) = \frac{1}{3}(a + 2d) \quad (2.29)$$

$$d_1 = \frac{1}{6}(E(y|M_1 M_1) - E(y|M_1 M_2)) = \frac{1}{6}(a + 2d) \quad (2.30)$$

Therefore, overall effect of of genotype $M_1 M_1$ in the B1 backcross is given by

$$b_{M_1 M_1} = a_1 + d_1 = \frac{1}{2}(a + 2d). \quad (2.31)$$

If all genotypes occurring in the F2 population do not occur in the study population, then it is not possible to separately estimate each F2 genotypic effect. For example, we cannot separately estimate (F2) additive and dominance effects using a backcross population. However, if combined data from both the B1 and the B2 backcross is used, then it is possible to separate the additive and dominance effects.

2.3.3 Partitioning the genetic variance

Equation (2.25) suggests the following linear model for an individual trait value y within the subpopulation having genotype x .

$$\begin{aligned} y_x &= E(y|x) + \varepsilon \\ &= \mu + G_x + \varepsilon, \end{aligned} \tag{2.32}$$

where G_x is the overall genetic effect and ε is a random error term having mean zero. This is the linear model of Cockerham (1954) and Kempthorne (1954) for an individual trait value. Following Fisher (1918), they used it to partition the total trait variance in terms of both the genetic variance (at a locus) and the variance due to error.

$$\begin{aligned} \text{var}(y) &= E(y^2) - E^2(y) \\ &= E E(y^2|x) - \mu^2 \\ &= E(\mu^2 + G_x^2 + \varepsilon^2 + 2\mu G_x + 2\mu\varepsilon + 2\varepsilon G_x) - \mu^2 \\ &= E(G_x^2) + 2E(\varepsilon G_x) + E(\varepsilon^2) \\ &= \text{var}(G_x) + 2\text{cov}(\varepsilon, G_x) + \text{var}(\varepsilon), \end{aligned} \tag{2.33}$$

since $E(G_x) = 0$ and $E(\varepsilon) = 0$ by assumption.

The part of the trait variance which is due to genetic effects is called the total genetic variance.

$$\text{total genetic variance} = \text{var}(G_x) + 2\text{cov}(\varepsilon, G_x) \tag{2.34}$$

If there are there are no interactions between genetic effects and other sources of variation such as environmental effects then.

$$\text{total genetic variance} = \text{var}(G_x) \quad (2.35)$$

If there is no covariance between different types of genetic effects (i.e. $\text{cov}(A_x, D_x) = 0$, $\text{cov}(A_x, (AD)_x) = 0$, and so on), then the genetic variance may be partitioned as follows

$$\text{var}(G_x) = \text{var}(A_x) + \text{var}(D_x) + \text{var}((AD)_x) + \text{var}((AA)_x) + \text{var}((DD)_x) \quad (2.36)$$

Furthermore, if there is no covariance between different types of genetic effects and each type of genetic effect has mean zero, then we also have the simplification given in Equation (2.37) below.

$$\text{var}(G_x) = E(A_x^2) + E(D_x^2) + E((AD)_x^2) + E((AA)_x^2) + E((DD)_x^2) \quad (2.37)$$

Equation (2.37) holds true for the Cockerham-Kempthorne model because the latter is based on orthogonal contrasts, which ensure zero covariance between different types of genetic effects.

In the Quantitative Genetics literature, proportion $H^2 = \text{var}(G_x)/\text{var}(y)$ is called the ‘broad sense heritability’ and the proportion $h^2 = \text{var}(A_x)/\text{var}(y)$ is called the ‘narrow sense heritability’. The broad sense heritability is the proportion of the trait variance that is explained by the total variability of the genetic effects, while the narrow sense heritability is the proportion of the trait variance that is explained by variability of the additive effects. The ‘narrow sense heritability’ is important in breeding programs because it is often associated with the degree of resemblance between relatives (Falconer and Mackay, 1996, page 123).

2.3.4 Number of genetic effects in a full linear regression model

Linear regression is commonly used to estimate genetic effects using marker and trait data. The maximum number of genetic effects that can be directly estimated by linear regression is one less than the number of distinct genotype groups. Any extra genetic effects may then be estimated from the fitted means.

Consider ℓ loci and suppose that, for the population under study, there are κ genotypes at each locus. Then there are κ^ℓ possible ℓ -locus genotypes. A full linear model includes the maximum of κ^ℓ effects. These effects are the intercept and $\kappa^\ell - 1$ genetic effects. The number of i -way genetic effects in a full model is equal to

$$\binom{\ell}{i}(\kappa - 1)^i \text{ where } i = 1, \dots, \ell.$$

There are $\ell(\kappa - 1)$ main effects and the number of interaction effects is equal to

$$\sum_{i=2}^{\ell} \binom{\ell}{i}(\kappa - 1)^i = \kappa^\ell - 1 - \ell(\kappa - 1).$$

This implies that the number of interaction terms can increase rapidly as the number of loci increases. Including a large number of effects in a model can adversely affect the resolution of point estimates. This because the sample size may not be large enough to permit accurate parameter estimation.

In the above illustration of the Cokerham-Kempthorne linear model (see Equation (2.25)), all terms of order three and above were ignored. In practice, such terms are often ignored in model fitting, not because they are insignificant but because of constraints imposed by small sample size. In fact many QTL mapping procedures do not estimate any of the epistatic effects. Rather than ignoring epistasis altogether it can be useful to fit a few low order interaction effects: for example, up to the second order as in Equation (2.25).

The next chapter (Chapter 3) examines the distribution of the trait for inbred line-cross populations. The distributions of sample means and sample variance of the trait are also examined. Later, the whole of Chapter 4 is devoted to regression methods for estimating the effects of QTL genotypes.

Chapter 3

The Inherent Mixture

This chapter gives a structural overview of the QTL mapping problem in the context of inbred line-crosses.

3.1 Statistical Exploration of Line-Cross data

Any population generated by an inbred line cross experiment has natural partitions, determined by groups of individuals having identical genotypes at certain loci. There are a large number of such partitions but our attention is restricted to distinct subgroups involving individuals who are genetically homogeneous at a specific set of marker loci. This restriction is unavoidable because the data provides information only for those markers at which individuals have been genotyped.

Suppose that n individuals have been genotyped at a fixed number of loci and that the experimental design yields s distinct marker genotypes. For each individual, the observed attributes are marker genotypes and one or more measurable traits of interest. For simplicity, assume that observations are made on a single trait. Classify the trait values according to the corresponding marker genotypes. This leads to a

view of the measurements as values of the random variables

$$\{Y_{ij} : i = 1, \dots, s; j = 1, \dots, n_i\},$$

where $n_i \geq 0$ is the number of sampled individuals having marker genotype i . Note that some marker genotypes may not appear in the sample. We denote the set of observed trait values by $\{y_{ij}\}$.

The goal of QTL analysis is to make inferences about QTL using available marker and trait information. It is therefore useful to consider also a hypothetical labelling of the trait data based on a partition determined by joint marker and QTL genotypes. Thus, an alternative representation of the trait data is as values of random variables $\Gamma_{ik\ell}$, where i indexes the marker genotype, k indexes the QTL genotype and ℓ indexes the individuals within genotype-class ik . Suppose that there are t possible QTL genotypes. The trait data can be denoted by $\{\gamma_{ik\ell} : i = 1, \dots, s; k = 1, \dots, t; \ell = 1, \dots, n_{ik}\}$, where n_{ik} is the number of sampled individuals having genotype ik . Essentially, the elements of the set $\{y_{ij}\}$ are rearranged to form the set $\{\gamma_{ik\ell}\}$ via an unobservable, one-to-one mapping.

Assume that the trait values are normally distributed (possibly after transformation) within genetically homogenous sub-populations. Assume that non-genetic sources of variation are completely random so that the $\Gamma_{ik\ell}$ are independently distributed as $N(\mu_{ik}, \sigma^2)$. The trait means, μ_{ik} , may vary for different genotypes. Typically, the common variance σ^2 encapsulates variability due to non-genetic factors as well as genetic factors that are not modelled.

Tables 3.1 and 3.2 summarize the properties of the population and sample described above. The population means given in Table 3.1 are functions of population genotype probabilities and genotypic effects (additive, dominance and interaction effects at and between loci). Marker and QTL position, generally given as pairwise recombination fractions, help to determine population genotype probabilities.

Table 3.1: Some population properties of line cross designs

Population Property	Notation and Comments	Known?
Probability of genotype ik	p_{ik} (a function of both the breeding design and unknown recombination probabilities)	No
Probability of marker genotype i	p_i (a function of the breeding design and known marker map)	Yes
Trait mean for genotype ik	$\mu_{ik} = E(\Gamma_{ik\ell} ik)$	No
Trait mean for marker-class i	$\mu_i = E(\Gamma_{ik\ell} i) = \sum_{k=1}^t \frac{p_{ik}}{p_i} \mu_{ik}$	No
Population mean trait value	$\mu = E(\Gamma_{ik\ell})$ $= \sum_{i=1}^s p_i \mu_i = \sum_{i=1}^s \sum_{k=1}^t p_{ik} \mu_{ik}$	No
Trait variance for genotype ik	$\sigma^2 = \sigma_{\text{error}}^2 = E(\Gamma_{ik\ell} - \mu_{ik})^2$ $= E(\Gamma_{ik\ell}^2) - \mu_{ik}^2$	No
Trait variance for marker-class i	$\sigma_i^2 = E((\Gamma_{ik\ell} - \mu_i)^2 i)$ $= E(\Gamma_{ik\ell}^2 i) - \mu_i^2$ $= \sum_{k=1}^t \frac{p_{ik}}{p_i} E(\Gamma_{ik\ell}^2) - \mu_i^2$ $= \sigma^2 + \sum_{k=1}^t \frac{p_{ik}}{p_i} \mu_{ik}^2 - \mu_i^2$	No
Overall trait variance	$\sigma_{\text{total}}^2 = E(\Gamma_{ik\ell} - \mu)^2 = E(\Gamma_{ik\ell}^2) - \mu^2$ $= \sum_{i=1}^s p_i E(\Gamma_{ik\ell}^2 i) - \mu^2$ $= \sigma^2 + \sum_{i=1}^s \sum_{k=1}^t p_{ik} \mu_{ik}^2 - \mu^2$	No

Table 3.2: Some sample properties of line cross designs

Sample Property	Notation and Comments	Known?
No. of marker genotypes	s (observed)	Yes
No. of QTL genotypes	t (fixed by assumption at the model specification stage)	Yes
Count for genotype ik	n_{ik} (unobservable)	No
Count for marker genotype i	$n_i = \sum_{k=1}^t n_{ik}$ (observed)	Yes
Total sample size	$n = \sum_{i=1}^s n_i$ (observed)	Yes
The ℓ^{th} trait value belonging to genotype-class ik , where $i = 1, \dots, s$; $k = 1, \dots, t$.	$\gamma_{ik\ell}$, $\ell = 1, \dots, n_{ik}$. (unobservable)	No
The j^{th} trait value belonging to marker-class i , where $i = 1, \dots, s$.	y_{ij} , $j = 1, \dots, n_i$	Yes
Sample mean for genotype-class ik	$\bar{\gamma}_{ik} = \frac{1}{n_{ik}} \sum_{\ell=1}^{n_{ik}} \gamma_{ik\ell}$	No
Sample mean trait value for marker-class i	$\bar{y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{ij}$	Yes
Overall sample mean	$\bar{y} = \frac{1}{n} \sum_{i=1}^s \sum_{j=1}^{n_i} y_{ij}$	Yes

Table 3.2: (continued)

Sample Property	Notation and Comments	Known?
Asymptotic distribution of the sample mean for marker-class i	$\bar{Y}_i \sim N\left(\frac{1}{n_i} \sum_{k=1}^t n_{ik} \mu_{ik}, \frac{\sigma_i^2}{n_i}\right)$	No
Asymptotic distribution of the overall sample mean	$\bar{Y} \sim N\left(\frac{1}{n} \sum_{i=1}^s \sum_{k=1}^t n_{ik} \mu_{ik}, \frac{\sigma_{\text{total}}^2}{n}\right)$	No
Sample variance of the trait values in genotype-class ik	$S_{ik}^2 = \frac{1}{(n_{ik}-1)} \sum_{k=1}^t \sum_{\ell=1}^{n_{ik}} (\gamma_{ik\ell} - \bar{\gamma}_{ik})^2$	No
Sample variance for marker-class i	$S_i^2 = \frac{1}{(n_i - 1)} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2$ $= \frac{n_i}{n_i - 1} \left(\frac{1}{n_i} \left(\sum_{j=1}^{n_i} y_{ij}^2 \right) - \bar{y}_i^2 \right)$ $= \frac{n_i}{n_i - 1} \left(\left(\sum_{k=1}^t \sum_{\ell=1}^{n_{ik}} \frac{\gamma_{ik\ell}^2}{n_i} \right) - \left(\sum_{k=1}^t \frac{n_{ik}}{n_i} \bar{\gamma}_{ik} \right)^2 \right)$	Yes
Asymptotic distribution of $\frac{(n_i - 1)}{\sigma^2}$ times the sample variance for marker-class i	<p>The statistic $\frac{(n_i - 1)S_i^2}{\sigma^2}$ is distributed as a non-central chi-square random variable, with $(n_i - 1)$ degrees of freedom and non-centrality parameter λ_i, where</p> $\lambda_i = \frac{n_i}{\sigma^2} \left(\sum_{k=1}^t \frac{n_{ik}}{n_i} \mu_{ik}^2 - \left(\sum_{k=1}^t \frac{n_{ik}}{n_i} \mu_{ik} \right)^2 \right).$	No

The sample mean for marker class i is asymptotically normally distributed, but the counts n_{i1}, \dots, n_{it} are not observed. We find all possible partitions (by QTL-type) within the i^{th} marker-class, and weight the resulting density of \bar{Y}_i by the conditional probability of such a partition, summing over the weighted densities. This gives the marginal distribution of \bar{Y}_i as a mixture of Normals.

The distribution of the sample variance within a given marker class must also be estimated by a mixture distribution. Although $\frac{(n_i-1)S_i^2}{\sigma^2}$ is asymptotically distributed as a noncentral chi-square with $n_i - 1$ degrees of freedom and non-centrality parameter λ_i as in Table 3.2, the counts n_{i1}, \dots, n_{it} are not observed. Behboodian (1972b) showed the distribution of the sample variance from such a mixed population is a multinomial mixture of non-central chi-squares.

3.2 From marker to QTL

The number and location of QTLs in the system are unknown. The aim of QTL mapping is first to detect the presence of QTL effect, by looking for statistical significance that can be attributed to QTL in a particular segment of the genome. The best case would be a situation in which significance is known to be attributable to genes within a particular marker interval. Where this case cannot be achieved, the possibility of attributing significance to the incorrect segment exists and can lead to the detection of “ghost” or false QTL.

If the detection is significant, then the next goal is to estimate the gene location in terms of the probability of recombinations between its locus and that of a nearby marker. Hence, an initial estimate of QTL location comes in terms of the recombination fractions between each QTL and a specific marker. Later, this is converted to *genetic distance* using a map function such as Haldane’s map function. Map functions are discussed in detail in the literature (see Ott 1991, pages 14-19 and pages 120-129;

Liu 1997, pages 318-329).

The conventional approach to QTL detection assumes a system containing a specific number and ordering of putative QTL linked to known markers. For any particular population and sampling design, the basic information comprises the marker linkage-map, the experimental design itself, the observed marker genotypes and the corresponding trait measurements taken for each sampled individual.

Properties of the linkage map and the experimental design are used together in estimating the conditional distribution of each QTL genotype given a marker genotype. Estimation of this distribution typically requires strong assumptions about the level and structure of recombinational interference between loci, the number of putative QTL, and the ordering of QTL relative to the markers. The estimated trait distribution, conditional on the observed marker and QTL genotypes, is used to make inferences about the sizes of QTL effects and the location(s) of the QTL(s) relative to the markers.

For convenience, denote the marker genotype and the QTL genotype by i , k respectively. If multiple marker loci are involved then i is a multi-locus genotype. Similarly, if we consider multiple QTL loci, then k is a multi-locus genotype. Also, suppose that

- y_{ij} is the trait value of individual ij , where individual ij is the j^{th} individual having marker genotype i ;
- m_{ij} is the marker genotype of individual ij ;
- q_{ij} is the QTL genotype of individual ij .

The foundation of QTL mapping theory is based on the two models listed below.

1. There must be a model, $w_{(ij)k}(\phi)$, for the probability that individual ij has QTL genotype k given that he/she has marker genotype i . Here ϕ is a vector of parameters controlling gene and genotype probabilities. Now, $m_{ij} = i$ by observation. However, there is uncertainty about q_{ij} . The uncertainty about q_{ij}

may be expressed as

$$q_{ij} = k, \text{ with probability } w_{(ij)k}(\phi).$$

2. There must be a model, $P(y_{ij}|m_{ij} = i, q_{ij} = k; \theta) = P(y_{ij}|i, k; \theta)$, for the conditional trait distribution given genotype ik . Here θ is a vector parameters which are thought to control phenotypic value. In genetic linkage studies, $P(y_{ij}|i, k; \theta)$ is often referred to as the *penetrance* of the trait

The conditional probability, $w_{(ij)k}(\phi)$, of being in QTL class k given membership of marker class i depends on the breeding design, the level of crossover interference and the linkage map (the positions of the markers and the QTL along the genome). Therefore, the parameter vector ϕ usually captures factors affecting gene and genotype probabilities within a population, such as population structure, gene transmission probabilities from parent to offspring and genotype by environmental interactions.

Let $M_{\text{sire}_{ij}}$, $M_{\text{dam}_{ij}}$ denote the marker genotype of the mother and father, respectively, of individual ij . Also, let $Q_{\text{sire}_{ij}}$, $Q_{\text{dam}_{ij}}$ denote the QTL genotype of the father (sire_{ij}) and mother (dam_{ij}), respectively, of individual ij . The conditional probability $w_{(ij)k}(\phi)$ is calculated by averaging over the possible parental QTL genotypes as in Equation (3.1).

$$\begin{aligned}
 & w_{(ij)k}(\phi) \\
 &= P(q_{ij} = k | m_{ij} = i; \phi, \text{ marker genotypes of parents } \text{sire}_{ij} \text{ and } \text{dam}_{ij}) \\
 &= \sum_{M_{\text{sire}_{ij}}} \sum_{M_{\text{dam}_{ij}}} \sum_{Q_{\text{sire}_{ij}}} \sum_{Q_{\text{dam}_{ij}}} \left(P(q_{ij} = k | m_{ij} = i, Q_{\text{sire}(ij)}, M_{\text{sire}_{ij}}, Q_{\text{dam}_{ij}}, M_{\text{dam}_{ij}}; \phi) \right. \\
 &\quad \times P(Q_{\text{sire}_{ij}} | M_{\text{sire}_{ij}}; \phi) P(M_{\text{sire}_{ij}}; m_{ij}, \phi) \\
 &\quad \times P(Q_{\text{dam}_{ij}} | M_{\text{dam}_{ij}}; \phi) P(M_{\text{dam}_{ij}}; m_{ij}, \phi) \left. \right) \quad (3.1)
 \end{aligned}$$

If the mother's marker genotype is known, then $P(M_{\text{dam}_{ij}}; m_{ij}, \phi) = 1$ for the corresponding observed marker genotype, and zero for all other marker genotypes. Likewise, if the father's marker genotype is known, then $P(M_{\text{sire}_{ij}}; m_{ij}, \phi) = 1$ for the observed paternal marker genotype, and $P(M_{\text{sire}_{ij}}; m_{ij}, \phi) = 0$ for all other marker genotypes. Therefore, if the parent marker genotypes are known, then Equation (3.1) reduces to

$$w_{(ij)k}(\phi) = \sum_{Q_{\text{sire}_{ij}}} \sum_{Q_{\text{dam}_{ij}}} \left(P(q_{ij} = k | m_{ij} = i, Q_{\text{sire}_{ij}}, M_{\text{sire}_{ij}}, Q_{\text{dam}_{ij}}, M_{\text{dam}_{ij}}; \phi) \right. \\ \left. \times P(Q_{\text{sire}_{ij}} | M_{\text{sire}_{ij}}; \phi) P(Q_{\text{dam}_{ij}} | M_{\text{dam}_{ij}}; \phi) \right) \quad (3.2)$$

Note that $w_{(ij)k}(\phi) \geq 0$ and $\sum_{k=1}^t w_{(ij)k}(\phi) = 1$.

Generally, the conditional probability $w_{(ij)k}(\phi)$ is easier to calculate for inbred designs than for outbred designs. This is mainly because, in simple inbred designs such as the backcross and the F2 intercross, each inbred parental population is assumed to be genetically homogenous at all QTL and marker loci. Therefore, if the observed data is from a single inbred population, we can drop the subscript j and write

$$w_{(ij)k}(\phi) = w_{ik}(\phi) = \frac{p_{ik}}{p_i},$$

where p_{ik} is the probability of genotype ik and p_i is the probability of marker genotype i in the inbred population. Also, relationships between siblings and other relatives, shared maternal effects, and other shared environmental effects are somewhat under experimental control for inbred line-cross designs. However, population structure, breeding and shared environmental effects are more difficult to control for outbred designs.

In the model, $P(y_{ij} | m_{ij} = i, q_{ij} = k; \theta)$, for the conditional trait distribution, the parameter vector θ can capture genetic effects as well as the effects of any extra covariates, cofactors and interactions that are assumed to affect trait value.

Marker-based methods depend on the marginal distribution of the trait value given the observed marker genotype. Using the theorems of conditional probability, we obtain the marginal trait distribution conditional on marker i as the finite mixture distribution displayed in Equation (3.3).

$$P(y_{ij} | m_{ij} = i; \boldsymbol{\phi}, \boldsymbol{\theta}) = \sum_{k=1}^t w_{(ij)k}(\boldsymbol{\phi}) P(y_{ij} | m_{ij} = i, q_{ij} = k; \boldsymbol{\theta}) \quad (3.3)$$

Assume that the trait is normally distributed within each genotype class ik , with common variance σ^2 but distinct means μ_{ik} in each class. If the assumption *neutral markers* is made, then the mean, μ_{ik} , depends only on the QTL genotype because the trait is assumed to be unaffected by the marker genotypes. We may write $\mu_{ik} = \mu_k$ under the neutral marker assumption. Individuals within each class ik are genetically homogeneous at the marker and QTL loci, therefore the within-class variance is assumed to be equal to the error variance, σ^2 . In the case of this Normal distribution, we have the Normal mixture density given in Equation (3.4) below.

$$f(y_{ij}; \boldsymbol{\phi}, \boldsymbol{\theta}) = f(y_{ij}; \boldsymbol{\phi}, \mu_{i1}, \dots, \mu_{it}, \sigma^2) = \sum_{k=1}^t \frac{w_{(ij)k}(\boldsymbol{\phi})}{\sigma \sqrt{2\pi}} \exp\left\{\frac{(y_{ij} - \mu_{ik})^2}{-2\sigma^2}\right\},$$

where $\boldsymbol{\theta} = (\mu_{i1}, \dots, \mu_{it}, \sigma^2)$. (3.4)

In this model, the error variance actually comprises within individual variation plus external environmental variation (see for example Falconer and Mackay 1996). More sophisticated models may explicitly include parameters for estimating the effects of one or more environmental factors.

The likelihood for a sample needs to take into account any relationships between relatives. This is achieved by considering possible values for the n -dimensional vector

$(q_{11}, \dots, q_{sn_s})$ containing QTL genotypes for all sample members (see Equation (3.5)).

$$\begin{aligned}
& L(y_{11}, \dots, y_{sn_s} | m_{11}, \dots, m_{sn_s}; \phi, \theta) \\
&= \sum_{q_{11}, \dots, q_{sn_s}} P(\mathbf{y} | m_{11}, \dots, m_{sn_s}, q_{11}, \dots, q_{sn_s}; \theta) P(q_{11}, \dots, q_{sn_s} | m_{11}, \dots, m_{sn_s}; \phi) \\
&= \sum_{q_{11}, \dots, q_{sn_s}} \prod_{i=1}^s \prod_{j=1}^{n_i} P(y_{ij} | m_{ij}, q_{ij}; \theta) P(q_{ij} | m_{ij}; \phi), \tag{3.5}
\end{aligned}$$

The last line of Equation (3.5) rests on three assumptions. These assumptions are:

1. The trait is genetically determined.
2. If an individual's phenotypic value is conditioned on his/her genotype, then its conditional distribution is independent of all other genotypes or phenotypic values in the pedigree.
3. If an individual's genotype is conditioned on the genotypes of his/her parents, then its conditional distribution is independent of all other individuals (except his/her parents).

If the parents of individual ij are in the sample, then $P(q_{ij} | m_{ij}; \phi)$ depends on genotypes of sire_{ij} , dam_{ij} (see Equation (3.1)) and so the genotypes of some sample members may not be independent.

The summation in Equation (3.5) is taken over an n -dimensional space. Therefore, the likelihood may be computationally demanding to calculate for large pedigrees. Several algorithms have been proposed in the literature to reduce the number of arithmetic operations. Examples include the Peeling Algorithm (Elston and Stewart, 1971; Cannings *et al.*, 1978) and the VITESSE Algorithm (O'Connell and Weeks, 1995).

For inbred designs, individuals are all from the same generation. Therefore, they are regarded as independent and the likelihood for a sample of such individuals has

the simple form given in Equation (3.6).

$$\begin{aligned}
& L(y_{11}, \dots, y_{sn_s} | m_{11}, \dots, m_{sn_s}; \boldsymbol{\phi}, \boldsymbol{\theta}) \\
&= \prod_{i=1}^s \prod_{j=1}^{n_i} \left(\sum_{k=1}^t P(q_{ij} | m_{ij} = i; \boldsymbol{\phi}) P(y_{ij} | m_{ij} = i, q_{ij} = k; \boldsymbol{\theta}) \right) \\
&= \prod_{i=1}^s \prod_{j=1}^{n_i} \left(\sum_{k=1}^t w_{(ij)k}(\boldsymbol{\phi}) P(y_{ij} | m_{ij} = i, q_{ij} = k; \boldsymbol{\theta}) \right) \tag{3.6}
\end{aligned}$$

In this thesis, we are focusing on inbred line-cross populations. Therefore, the form of the likelihood given in Equation (3.6) is of primary interest.

For complete specification of the probability densities, parameter estimates are needed for $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$. Knowledge of $\boldsymbol{\phi}$ allows calculation of recombination fractions between marker loci and QTL. Knowledge of the genotypic means, μ_{ik} allows calculation of genotypic effects, which is the first step in determining whether genes exist that significantly affect trait value. Maximum likelihood estimation and marker-trait regressions are among the most common methods for estimating these parameters. In Chapter 4, we discuss QTL mapping by multiple regression and in Chapter 5, we explore the mixture-likelihood approach to QTL mapping.

Chapter 4

Regression Methods

Fixed effects linear models have been used, with moderate success, to detect marker-trait associations and to estimate QTL effects (see for example Haley and Knott, 1992; Martinez and Curnow, 1992; Whittaker *et al.*, 1996). This mapping technique detects QTL by relating the regression of trait on marker-genotype to a regression of trait on putative QTL-genotypes. The fixed effects regression or Analysis of variance (ANOVA) models may be implemented using a variety of constraints and contrasts/coding definitions. The purpose of this chapter is to consolidate and formalize the existing strategies for inferring QTL from multiple regressions of trait value on marker genotype.

Section 4.1 introduces the theory and notation of regression in the QTL mapping context. An example follows in Section 4.2, where the theory is applied to the F₂, and the reader is encouraged to compare the general results in the notation-heavy Section 4.1 with the specific realisation presented in Section 4.2.

4.1 Multiple Regression with Contrasts

4.1.1 Models, contrasts and implications

Consider a system of s distinct, possibly multi-locus, marker genotypes. The number of distinct marker genotypes, s , depends on the experimental population, the number of marker loci, and on whether or not the loci are codominant. For example, in the case of two codominant markers, $s = 4$ for the backcross design, and $s = 9$ for the F2 design. In the case of two dominant marker loci, heterozygous genotypes cannot be distinguished from homozygous genotypes, so $s = 1$ for the backcross design, and $s = 4$ for the F2 design.

Given n individuals, the simplest ANOVA model considers the trait value, y_{ij} , of the j^{th} individual with marker genotype i as a function of the background effects u_0 , marker effect u_i and a random error ε_{ij} .

$$y_{ij} = u_0 + u_i + \varepsilon_{ij}. \quad (4.1)$$

The error terms $\{\varepsilon_{ij}\}$ are assumed to be independent, identically distributed normal random variables having unknown variance (denoted by σ^2) and mean zero.

Define the mean trait-value of an individual having marker genotype i as

$$\mu_i = E(Y | \text{genotype } i) = E(Y|i). \quad (4.2)$$

Then

$$\mu_i = u_0 + u_i. \quad (4.3)$$

Let $\boldsymbol{\mu} = (\mu_1, \mu_2, \dots, \mu_s)^T$ and let $\mathbf{p} = (p_1, p_2, \dots, p_s)^T$, where $\sum_{i=1}^s p_i = 1$ and p_i is the probability of marker genotype i within the population from which the sample was drawn. The characteristics of this population depend on the breeding design. Let μ be the population mean trait-value. Then

$$\mu = \mathbf{p}^T \boldsymbol{\mu} = u_0 + \sum_{i=1}^s p_i u_i. \quad (4.4)$$

The model is over-parameterised and requires a constraint on the u_i . Various constraints are possible, and if (for example) we set $\sum_{i=1}^s p_i u_i = 0$ then the constant term u_0 is equal to the overall mean μ .

Label the individuals so that individual ij is the j^{th} individual having marker genotype i . Now let $m_{(ij)i'}$ be a binary indicator variable for the marker genotype of individual ij . Then

$$m_{(ij)i'} = \delta_{ii'} = \begin{cases} 1 & \text{if } i = i' \\ 0 & \text{otherwise.} \end{cases}$$

Equation (4.1) represents a one-way ANOVA model. Equivalently, we may express this model in the form of a multiple regression, with the phenotypic value for the j^{th} individual in marker category i given by

$$y_{ij} = u_0 + \sum_{i'=1}^s u_{i'} m_{(ij)i'} + \varepsilon_{ij}, \quad (4.5)$$

Define the matrix $\mathbf{M}_{n \times s} = (m_{(ij)i'})$, a binary incidence matrix where each row has the value one in the column for the genotype of the corresponding individual, and zero in all other columns. As in Table 3.2, denote the number of individuals having marker genotype i by n_i . Also let $\mathbf{y} = (y_{11}, y_{12}, \dots, y_{sn_s})^T$ be a vector of trait values; $\mathbf{1}_n$ a column vector of order n with each element equal to one; $\mathbf{u} = (u_0, u_1, u_2, \dots, u_s)^T$ and $\boldsymbol{\varepsilon} = (\varepsilon_{11}, \varepsilon_{12}, \dots, \varepsilon_{sn_s})^T$ be, respectively, vectors containing regression coefficients and independent, identically distributed error terms. Assume that the error terms have a Normal distribution with mean zero and unknown variance. Then the regression model may be written in matrix notation as follows:

$$\mathbf{y} = [\mathbf{1}_n \quad \mathbf{M}] \mathbf{u} + \boldsymbol{\varepsilon}, \quad (4.6)$$

The model cannot be fitted as given in Equation (4.6) because the model matrix is $\mathbf{X} = [\mathbf{1}_n \quad \mathbf{M}]$ which has $(s+1)$ columns and rank s , indicating redundancy in the model. That is, $\mathbf{X}^T \mathbf{X}$ does not have a left inverse, therefore a solution to the least

squares normalizing equations cannot be found. There are several ways to address the problem of redundancy in the model matrix. The simplest approach is to remove the constant term from the model. An alternative approach is to fit the model using the intercept together with a set of not more than $s - 1$ linearly independent vectors in \mathbb{R}^s .

In the context of linear models involving s regressors, any set of linearly independent vectors in \mathbb{R}^s is referred to as a set of contrast vectors. There is usually a constraint that elements of each contrast vector sum to zero (see for example Hochberg and Tamhane, 1987; Robertson *et al.*, 1988; Montgomery, 1996; Venables and Ripley, 1997). If the original model matrix is a $(n \times s)$ binary incidence matrix, and \mathbf{C} is any matrix whose columns are a set of linearly independent vectors in \mathbb{R}^s , then \mathbf{C} is a contrast generator in the sense that its left inverse forms a contrast matrix. When post-multiplied by the vector of treatment means, the rows of the matrix given by the left inverse of the contrast generator (\mathbf{C}) produces contrasts between treatment means. Each contrast generator has a unique left inverse. Therefore, in this thesis, it is convenient to use the term ‘contrast matrix’ rather loosely to when referring to the contrast generating matrices as well as when referring to the generated contrasts themselves.

To fit the model given in Equation (4.6) a matrix, $\mathbf{C}_{s \times (s-1)} = (c_{ip})$, having rank $(s - 1)$, is chosen so that \mathbf{X} has rank s where \mathbf{X} is the recoded model matrix given in Equation (4.7).

$$\mathbf{X} = [\mathbf{1}_n \quad \mathbf{MC}] \quad (4.7)$$

The linear model may now be written as

$$\mathbf{y} = \mathbf{X} \mathbf{b} + \boldsymbol{\varepsilon} = [\mathbf{1}_n \quad \mathbf{MC}] \mathbf{b} + \boldsymbol{\varepsilon}. \quad (4.8)$$

The solution by least squares is

$$\hat{\mathbf{b}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (4.9)$$

In addition to removing redundancy, fitting with contrasts yields the benefit of parameter estimates that can easily be interpreted as linear functions of treatment means. This facilitates hypothesis tests involving comparisons of treatment means. In QTL analysis, for example, we are concerned about differences between treatment means, where the ‘treatments’ are marker-genotype classes. Appropriately selected contrasts can be used to extract this information.

By comparing equations (4.6) and (4.8), we see that the relationship between the coefficient vectors \mathbf{u} and \mathbf{b} is

$$\begin{aligned} u_0 &= b_0 \text{ and } \mathbf{u}_1 = \mathbf{C} \mathbf{b}_1, \\ \text{where } \mathbf{u} &= (u_0, u_1, \dots, u_s)^T = (u_0, \mathbf{u}_1^T)^T \\ \text{and } \mathbf{b} &= (b_0, b_1, \dots, b_{s-1})^T = (b_0, \mathbf{b}_1^T)^T. \end{aligned} \quad (4.10)$$

If $\mathbf{a} = (a_1, a_2, \dots, a_s)^T$ is a vector such that $\mathbf{a}^T \mathbf{C} = \mathbf{0}$ then using \mathbf{b} as the vector of parameters amounts to estimation of the original parameters, \mathbf{u} , under the identification constraint $\mathbf{a}^T \mathbf{u}_1 = \mathbf{a}^T \mathbf{C} \mathbf{b}_1 = \mathbf{0}$, that is

$$\sum_{k=1}^s a_k u_k = 0.$$

Expanding the matrix structures in Equation (4.10) reveals that the components of \mathbf{b}_1 are related such that

$$b_p = \frac{\sum_{i=1}^s n_i c_{ip} u_i - \sum_{r \neq p} b_r \left(\sum_{i=1}^s n_i c_{ip} c_{ir} \right)}{\sum_{i=1}^s n_i c_{ip}^2} \text{ for } p = 1, \dots, s-1. \quad (4.11)$$

Let \mathbf{C}^+ denote the unique left inverse of \mathbf{C} .

$$\mathbf{C}^+ = (\mathbf{C}^T \mathbf{C})^{-1} \mathbf{C}^T \quad (4.12)$$

The new parameters $\mathbf{b} = (b_0, b_1, \dots, b_{s-1})$ may be interpreted in terms of the original parameters as follows:

$$b_0 = u_0 \text{ and } \mathbf{b}_1 = \mathbf{C}^+ \mathbf{u}_1. \quad (4.13)$$

Denote the p^{th} row of \mathbf{C}^+ by $\mathbf{C}_{p\bullet}^+$. Then we see that \mathbf{C} generates $(s - 1)$ contrasts of the form $\mathbf{C}_{p\bullet}^+ \mathbf{u}_1$, associated with the hypothesis or linear constraint $\mathbf{C}_{p\bullet}^+ \mathbf{u}_1 = 0$ where $p = 1, \dots, s - 1$.

Define

$$\mathbf{C}_m = [\mathbf{1}_s \quad \mathbf{C}] \quad (4.14)$$

then by the definition of \mathbf{M} , we have that $\mathbf{M}\mathbf{1}_s = \mathbf{1}_n$ and so

$$[\mathbf{1}_n \quad \mathbf{MC}] = \mathbf{MC}_m. \quad (4.15)$$

Therefore, Equation (4.8) is equivalent to Equation (4.16) below.

$$\mathbf{y} = \mathbf{MC}_m \mathbf{b} + \boldsymbol{\varepsilon} \quad (4.16)$$

The estimated mean trait values for the marker categories are given by

$$\hat{\boldsymbol{\mu}} = \mathbf{C}_m \hat{\mathbf{b}}. \quad (4.17)$$

When discussing various features of the model, sometimes it is easier use \mathbf{C}_m and sometimes it is easier use \mathbf{C} .

For convenience, let $\mathbf{C}_{\bullet p}$ denote the p^{th} column of the matrix \mathbf{C} . If $\mathbf{X} = \mathbf{MC}_m$ is an orthogonal model matrix, then the columns of \mathbf{C} are called orthogonal contrasts. As we shall see later, the use of orthogonal contrasts ensures that the least squares estimates of the parameters $\{b_p\}$ are uncorrelated. This results in a convenient partitioning of the regression sums of squares. The matrix \mathbf{C} is an orthogonal contrast matrix if $\mathbf{X}^T \mathbf{X}$ is diagonal, where \mathbf{X} is the model matrix defined in Equation (4.7). For a model of the form given in Equation (4.8) we have

$$\begin{aligned} \mathbf{X}^T \mathbf{X} &= \begin{bmatrix} \mathbf{1}_n^T \\ \mathbf{C}^T \mathbf{M}^T \end{bmatrix} [\mathbf{1}_n \quad \mathbf{MC}] \\ &= \begin{pmatrix} n & \mathbf{1}_n^T \mathbf{MC} \\ (\mathbf{MC})^T \mathbf{1}_n & (\mathbf{MC})^T \mathbf{MC} \end{pmatrix}. \end{aligned} \quad (4.18)$$

In order for $\mathbf{X}^T \mathbf{X}$ to be diagonal, we must have $\mathbf{1}_n^T \mathbf{M} \mathbf{C} = \mathbf{0}$ and $(\mathbf{M} \mathbf{C})^T \mathbf{M} \mathbf{C}$ must be a diagonal matrix. Therefore, for a regression model involving the constant term, any two distinct contrasts $\mathbf{C}_{\bullet p} = (c_{1p}, \dots, c_{sp})^T$ and $\mathbf{C}_{\bullet r} = (c_{1r}, \dots, c_{sr})^T$ (with $p \neq r$) are orthogonal if $\mathbf{C}_{\bullet p}^T \mathbf{M}^T \mathbf{M} \mathbf{C}_{\bullet r} = 0$, $\mathbf{1}_n^T \mathbf{M} \mathbf{C}_{\bullet p} = 0$ and $\mathbf{1}_n^T \mathbf{M} \mathbf{C}_{\bullet r} = 0$. Note that $\mathbf{1}_n^T \mathbf{M} = (n_1, n_1, \dots, n_s)$ and that $\mathbf{M}^T \mathbf{M}$ is a diagonal matrix with n_i being the i^{th} diagonal element. Consequently, $\mathbf{C}_{\bullet p}$ and $\mathbf{C}_{\bullet r}$, where $p \neq r$, are orthogonal contrast vectors if

$$\sum_{i=1}^s n_i c_{ip} c_{ir} = 0 \quad (4.19)$$

$$\sum_{i=1}^s n_i c_{ip} = 0 \quad (4.20)$$

$$\sum_{i=1}^s n_i c_{ir} = 0. \quad (4.21)$$

If Equation (4.21) holds for all columns of \mathbf{C} , then the contrast vector $\mathbf{1}_s$ (which corresponds to the constant term) is orthogonal to every contrast in \mathbf{C} .

Taking Equation (4.10) together with the requirement that $\mathbf{1}_n^T \mathbf{M} \mathbf{C} = \mathbf{0}$ and the requirement that $(\mathbf{M} \mathbf{C})^T \mathbf{M} \mathbf{C}$ be diagonal implies the following:

1. When \mathbf{C} is an orthogonal contrast matrix, using it to estimate \mathbf{b} amounts to estimating the original parameters \mathbf{u} under the constraint that $\mathbf{1}_n^T \mathbf{M} \mathbf{u}_1 = \mathbf{0}$, giving $\sum_{i=1}^s n_i u_i = 0$. This is equivalent to the constraint that $\sum_{i=1}^s \hat{p}_i u_i = 0$, where $\hat{p}_i = n_i/n$.
2. When \mathbf{C} is an orthogonal contrast matrix, the components of \mathbf{b} satisfy the equations

$$b_0 = u_0 = \mu, \text{ and } b_p = \frac{\sum_{i=1}^s n_i c_{ip} u_i}{\sum_{i=1}^s n_i c_{ip}^2}, \text{ for } p = 1, \dots, s-1. \quad (4.22)$$

4.1.2 Sums of Squares and Hypothesis Tests

Consider the minimal (single-parameter) model given in equation (4.23). To test whether at least one component of \mathbf{b}_1 is non-zero, we compare the model given in Equation (4.8) to the minimal model which fits only the intercept term.

$$\mathbf{y} = \mathbf{1}_n b_0 + \boldsymbol{\varepsilon}. \quad (4.23)$$

For the minimal model, the maximum likelihood estimator of b_0 is $\hat{b}_0 = \bar{y}$.

Let $\bar{\mathbf{y}} = \mathbf{1}_n \bar{y}$, then the sum of squares from the ‘minimal’ model is given by

$$SS_{\text{total}} = (\mathbf{y} - \bar{\mathbf{y}})^T (\mathbf{y} - \bar{\mathbf{y}}) = \sum_{i=1}^s \sum_{j=1}^{n_i} (y_{ij} - \bar{y})^2. \quad (4.24)$$

To see how the constant term (b_0) of the minimal model becomes modified when the full model is fitted, it is useful to define a centered matrix \mathbf{A} as follows:

$$\mathbf{A} = \mathbf{M}\mathbf{C} - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T \mathbf{M}\mathbf{C} = \mathbf{H}\mathbf{M}\mathbf{C} \quad (4.25)$$

where

$$\mathbf{H} = \mathbf{I}_n - \frac{1}{n} \mathbf{1}_n \mathbf{1}_n^T. \quad (4.26)$$

Consider the full model given in Equation (4.8), its least squares solution given in Equation (4.9) and the partitioned matrix $(\mathbf{X}^T \mathbf{X})$ given in Equation (4.18). Applying the formulae for the inverse of a partitioned matrix allows us to partition the least squares solution to show separate expressions for \hat{b}_0 and $\hat{\mathbf{b}}_1$ (see Mardia *et al.*, 1979, pages 458-459; Rao and Toutenburg, 1995, pages 38-39).

$$\hat{\mathbf{b}} = \begin{pmatrix} \hat{b}_0 \\ \hat{\mathbf{b}}_1 \end{pmatrix} = \begin{pmatrix} \bar{y} - \frac{1}{n} \mathbf{1}_n^T \mathbf{M}\mathbf{C} (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \\ (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \end{pmatrix} = \begin{pmatrix} \bar{y} - \frac{1}{n} \mathbf{1}_n^T \mathbf{M}\mathbf{C} \hat{\mathbf{b}}_1 \\ (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{y} \end{pmatrix} \quad (4.27)$$

The fitted values may be written as

$$\hat{\mathbf{y}} = \mathbf{X} \hat{\mathbf{b}} = [\mathbf{1}_n \quad \mathbf{M}\mathbf{C}] \hat{\mathbf{b}} = \bar{\mathbf{y}} + \mathbf{A} \hat{\mathbf{b}}_1 \quad (4.28)$$

The fact that centering matrix \mathbf{H} is idempotent ($\mathbf{H}^T = \mathbf{H}$ and $\mathbf{H}\mathbf{H} = \mathbf{H}$) and the fact that $\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is also idempotent leads to the relationship given in Equation (4.29).

$$\begin{aligned}
 (\mathbf{y} - \bar{\mathbf{y}})^T \mathbf{A} \hat{\mathbf{b}}_1 &= \mathbf{y}^T \mathbf{H}^T \mathbf{A} \hat{\mathbf{b}}_1 \quad \text{by the definition of } \mathbf{H} \\
 &= \mathbf{y}^T \mathbf{A} \hat{\mathbf{b}}_1 \quad \text{because } \mathbf{H} \text{ is idempotent and } \mathbf{A} = \mathbf{H}\mathbf{M}\mathbf{C} \\
 &= \mathbf{y}^T \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \mathbf{y} \quad \text{by the definition of } \hat{\mathbf{b}}_1 \\
 &= \hat{\mathbf{b}}_1^T (\mathbf{A}^T\mathbf{A}) \hat{\mathbf{b}}_1 \quad \text{because } \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \text{ is idempotent.} \quad (4.29)
 \end{aligned}$$

The relationship given in Equation (4.29) helps us to relate the residual sum of squares to the terms of the total sum of squares. The residual sum of squares (SS_{error}) for the full model is therefore

$$\begin{aligned}
 SS_{\text{error}} &= (\mathbf{y} - \mathbf{X}\hat{\mathbf{b}})^T (\mathbf{y} - \mathbf{X}\hat{\mathbf{b}}) \\
 &= (\mathbf{y} - \bar{\mathbf{y}} - \mathbf{A}\hat{\mathbf{b}}_1)^T (\mathbf{y} - \bar{\mathbf{y}} - \mathbf{A}\hat{\mathbf{b}}_1) \\
 &= (\mathbf{y} - \bar{\mathbf{y}})^T (\mathbf{y} - \bar{\mathbf{y}}) - \hat{\mathbf{b}}_1^T (\mathbf{A}^T\mathbf{A}) \hat{\mathbf{b}}_1 \quad (4.30)
 \end{aligned}$$

The amount of variability explained by the regression, SS_{reg} (the regression sum of squares) is calculated by subtraction:

$$SS_{\text{reg}} = SS_{\text{total}} - SS_{\text{error}} = \hat{\mathbf{b}}_1^T (\mathbf{A}^T\mathbf{A}) \hat{\mathbf{b}}_1. \quad (4.31)$$

The regression sums of squares, SS_{reg} , has $(s - 1)$ degrees of freedom associated with it. Table 4.1, below, summarizes the sources of variation provided by the full model.

The variance of the vector of regression coefficients is given by

$$\text{var}(\hat{\mathbf{b}}) = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}. \quad (4.32)$$

The error (environmental) variance, σ^2 , is unknown so we estimate it by using the residual mean square.

$$\hat{\sigma}^2 = SS_{\text{error}} / (n - s) \quad (4.33)$$

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square
Regression on $\mathbf{MC}_{\bullet 1}, \mathbf{MC}_{\bullet 2}, \dots, \mathbf{MC}_{\bullet (s-1)}$	SS_{reg}	$s - 1$	$SS_{\text{reg}}/(s - 1)$
Residual error	SS_{error}	$(n - 1) - (s - 1)$	$SS_{\text{error}}/(n - s)$
Total	SS_{total}	$n - 1$	

Table 4.1: One-way ANOVA table.

The test for significant evidence for the truth of all contrasts ($H_0 : \hat{\mathbf{b}}_1 = \mathbf{0}$ versus $H_0 : \hat{\mathbf{b}}_1 \neq \mathbf{0}$) is based on the statistic

$$F_{\text{reg}} = \frac{SS_{\text{reg}}/(s - 1)}{SS_{\text{error}}/(n - s)} \sim F_{s-1, n-s} \quad (4.34)$$

This statistic is distributed according to the F -distribution with $(s - 1)$ and $(n - s)$ degrees of freedom, provided that the errors are independent and identically distributed with zero mean. If the F test given by Equation (4.34) is statistically significant, then there is evidence for genetic effects on the trait.

Each contrast has one degree of freedom associated with it. A test for $b_i = 0$ versus $b_i \neq 0$, is a Wald t -test based on regression estimators from the full model. The test statistic is

$$t_p = \frac{\hat{b}_p}{\sqrt{(\mathbf{X}^T \mathbf{X})_{pp}^{-1} \frac{SS_{\text{error}}}{(n-s)}}} \quad (4.35)$$

and it has an asymptotic t -distribution with $n - s$ degrees of freedom.

A test for inclusion of a subset containing $q < (s - 1)$ of the contrasts may be constructed by fitting reduced model which excludes this subset, and then comparing the reduced model to the full model via an F test.

$$F_p = \frac{(SS_{\text{reg}(\text{full})} - SS_{\text{reg}(\text{reduced})})/(s - q - 1)}{SS_{\text{error}}/(n - s)} \sim F_{s-q-1, n-s} \quad (4.36)$$

If the contrasts are not orthogonal, and we wish to test any subset of the contrasts, it is necessary to refit the entire model. On the other hand, using orthogonal contrasts

leads to parameter estimates that are independent. Therefore, we can add new terms to (or delete terms from) any sub-model without recomputing the $\{b_p\}$ already in (or remaining in) the model. When \mathbf{C} is an orthogonal contrast matrix, the estimators for regression coefficients have the form:

$$\hat{b}_0 = \bar{y}, \text{ and } \hat{\mathbf{b}}_1 = \text{diag} \left(\frac{1}{(\mathbf{A}_{\bullet 1})^T \mathbf{A}_{\bullet 1}}, \dots, \frac{1}{(\mathbf{A}_{\bullet (s-1)})^T \mathbf{A}_{\bullet (s-1)}} \right) \mathbf{A}^T \mathbf{y}, \quad (4.37)$$

where $\mathbf{A}_{\bullet p}$ is the p^{th} column of \mathbf{A} and

$$\mathbf{A}_{\bullet p} = \mathbf{HMC}_{\bullet p}. \quad (4.38)$$

For any contrast matrix \mathbf{C} , the regression sum of squares associated with fitting only the p^{th} contrast is given by

$$SS_{\text{reg}(p)} = \tilde{b}_p^T (\mathbf{A}_{\bullet p}^T \mathbf{A}_{\bullet p}) \tilde{b}_p, \text{ where } \tilde{b}_p = (\mathbf{A}_{\bullet p}^T \mathbf{A}_{\bullet p})^{-1} \mathbf{A}_{\bullet p}^T \mathbf{y}. \quad (4.39)$$

Orthogonal contrasts yield $\hat{b}_p = \tilde{b}_p$ and $SS_{\text{reg}} = \sum_{p=1}^{s-1} SS_{\text{reg}(p)}$ so the contrast sums of squares partition the regression sum of squares for the full model.

4.1.3 Inferring QTL from marker regression

The QTL are unknown, therefore standard regression models cannot explicitly include QTL genotype-indicators as explanatory variables. However, it is important to note that standard regression of trait-values on marker genotype-indicators does not model any information about recombination between marker and QTL.

Inference about QTL is made possible by establishing a linear relationship between an estimable subset of QTL effects and the estimated marker effects. This relationship is obtained by invoking the theorem of conditional probability while simultaneously making strong assumptions about

- (a) the number of QTL and the genotypes that they generate,

- (b) the recombination between loci and how recombination determines the conditional probabilities of QTL genotypes given marker genotypes (for the breeding design being studied).

The QTL effects are estimated from the fitted \mathbf{b} , and the process of doing so is equivalent to fitting a further model to the data. Details of this process are given in the remainder of this section and examples of contrast matrices are given in Section 4.1.4. Then, an example of this process is given in Section 4.2. Bullet points one to six, below, describe the basic steps for establishing a useable linear relationship between marker effects and QTL effects.

1. Assume a fixed number of QTL and suppose that, for the breeding design being studied, they generate t distinct QTL genotypes. If more than one locus is assumed to affect the trait, then the QTL genotypes are multi-locus genotypes. For example, a backcross model with one QTL will have $t = 2$ QTL genotypes.
2. Define $\boldsymbol{\mu}_q = (\mu_{q1}, \mu_{q2}, \dots, \mu_{qt})^T$, where μ_{qk} is the mean trait value for individuals having the k^{th} QTL genotype. All of the μ_{qk} are unknown parameters.
3. Introduce a matrix $\mathbf{W}_{s \times t} = (w_{ik})$, where w_{ik} is the conditional probability of the k^{th} QTL genotype given the i^{th} marker genotype. The conditional probabilities, w_{ik} , are generally non-linear functions of unknown recombination fractions.
4. Apply the theorem of conditional probability to obtain the relationship

$$\boldsymbol{\mu} = \mathbf{W}\boldsymbol{\mu}_q \quad (4.40)$$

between the marker-group means and the QTL-group means.

5. Estimates of the marker group means are obtained from the marker regression as $\hat{\boldsymbol{\mu}} = \mathbf{C}_m \hat{\mathbf{b}}$, so we have

$$\mathbf{C}_m \hat{\mathbf{b}} = [\mathbf{1}_s \quad \mathbf{C}] \hat{\mathbf{b}} = \mathbf{W}\boldsymbol{\mu}_q. \quad (4.41)$$

However, there is no guarantee that the matrix of conditional genotype probabilities, \mathbf{W} , has a left inverse. It does not have a left inverse if $t > s$ and even when $t \leq s$, the matrix \mathbf{W} could turn out to have linearly dependent columns. Unlike contrast matrices, the matrix \mathbf{W} is not a construction whose components we can change at will, rather it is determined by both the breeding design and the locations of the QTL. Lack of linear independence in the columns of \mathbf{W} typically occurs in multi-QTL models, making it impossible to completely separate the effects of different QTL. In order to reduce Equation (4.41) to a linear system of equations based on estimable functions of the QTL effects, we introduce (below) a QTL-contrast matrix to reduce the dimensionality (if necessary) and to facilitate comparison between QTL group means.

6. Introduce a QTL-contrast matrix, \mathbf{C}_q , having t rows and $t' \leq \min(s, t)$ columns, constructed such that

$$\text{rank}(\mathbf{W}\mathbf{C}_q) = \text{rank}(\mathbf{C}_q) = t' \text{ and } \mathbf{C}_q = [\mathbf{1}_t \quad \check{\mathbf{C}}]. \quad (4.42)$$

The columns of \mathbf{C}_q are associated with a column vector, \mathbf{b}_q , containing t' QTL regression coefficients, and with an unobserved binary matrix, $\mathbf{Z}_{n \times t}$, of indicators for QTL-genotype. The set of effects in \mathbf{b}_q is an estimable subset of QTL effects. Now we have the definition

$$\boldsymbol{\mu}_q = \mathbf{C}_q \mathbf{b}_q. \quad (4.43)$$

To place a meaningful interpretation on the regression coefficients \mathbf{b}_q , we note that \mathbf{b}_q is in fact the linear combination of QTL genotypic means given by

$$\mathbf{b}_q = (\mathbf{C}_q)^+ \boldsymbol{\mu}_q. \quad (4.44)$$

The desired linear relationship between the marker effects and the QTL effects is

$$\hat{\boldsymbol{\mu}} = \mathbf{C}_m \hat{\mathbf{b}} = \mathbf{W}\mathbf{C}_q \mathbf{b}_q. \quad (4.45)$$

The matrices \mathbf{C}_m , and \mathbf{C}_q are known constants, while the vector $\hat{\mathbf{b}}$ contains known estimates obtained from marker regression. Therefore, \mathbf{W} and \mathbf{b}_q are the only unknowns in Equation (4.45). The aim is to estimate any unknown parameters (recombination fractions) in \mathbf{W} and also to estimate \mathbf{b}_q (the QTL regression coefficients - which are linear functions of QTL effects).

The form of the solution is easy to obtain, but as we will see below, the solution may not be unique. Depending on the number of effects, the number of unknown recombination fractions and the configuration of \mathbf{W} , some systems may even generate infinitely many solutions because the components of \mathbf{b}_q may not be separable from the unknown recombination fractions. Moreover, the requirement that the rank of \mathbf{WC}_q be equal to its number of columns, implies that if $t' < t$, then some QTL effects may not be separable from each other.

Let us examine the form of the solution and features of the linear relationship between \mathbf{b} and \mathbf{b}_q which may be useful for testing hypotheses about QTL.

By definition, the matrices \mathbf{C}_m and \mathbf{WC}_q are of full column rank, so the left inverse exists in both cases. Multiplying Equation (4.45) by the unique left inverse of \mathbf{C}_m , yields a system of s equations given by

$$\hat{\mathbf{b}} = (\mathbf{C}_m)^+ \mathbf{WC}_q \mathbf{b}_q. \quad (4.46)$$

By assumption, all solutions $\{\hat{\mathbf{b}}_q, \widehat{\mathbf{W}}\}$ satisfy the relationship given in Equation (4.47).

$$\hat{\mathbf{b}}_q = (\widehat{\mathbf{W}}\mathbf{C}_q)^+ \mathbf{C}_m \hat{\mathbf{b}}. \quad (4.47)$$

Equations (4.16) and (4.46) suggest that if \mathbf{W} were known, then \mathbf{b}_q could have been directly estimated via a regression of trait value on marker genotype based on the model

$$\mathbf{y} = \mathbf{MC}_w \mathbf{b}_q + \boldsymbol{\varepsilon}, \quad (4.48)$$

where \mathbf{C}_w is a new contrast matrix defined as

$$\mathbf{C}_w = \mathbf{C}_m(\mathbf{C}_m)^+ \mathbf{W}\mathbf{C}_q. \quad (4.49)$$

If we treat \mathbf{W} as a known constant, then Equation (4.47) implies that the covariance matrices of the estimated effects $\widehat{\mathbf{b}}_q$ and $\widehat{\mathbf{b}}$ may be related as follows:

$$\text{var}(\widehat{\mathbf{b}}_q) = \mathbf{K} \text{var}(\widehat{\mathbf{b}}) \mathbf{K}^T, \text{ where } \mathbf{K} = (\mathbf{W}\mathbf{C}_q)^+ \mathbf{C}_m. \quad (4.50)$$

Haley and Knott (1992) and Martinez and Curnow (1992) independently proposed searching over several putative QTL locations by selecting recombination fractions from a grid (thus fixing \mathbf{W}), and then fitting a model equivalent to Equation (4.48), repeating the regression for each set of QTL locations. They suggested that the solution $\{\widehat{\mathbf{b}}_q, \widehat{\mathbf{W}}\}$ is the set of points that minimizes the residual sum of squares. It is important to note that the parameter estimates generated by this search procedure cannot be uniquely optimal unless Equation (4.46) actually has a unique solution.

If Equation (4.46) is such that the number of unknown parameters is greater than the number of equations (s), then one could assign valid, arbitrarily chosen values to inestimable parameters in \mathbf{b}_q and/or \mathbf{W} , thereby generating infinitely many solutions.

If, on the other hand, the number of unknown parameters is less than or equal to the number of equations, then a finite number of solutions exist. When a finite number of solutions exist, they may be found by solving the first t' equations generated by Equation (4.46) to obtain \mathbf{b}_q in terms of \mathbf{W} , and then back-substituting into the remaining $s - t'$ equations to find the unknown recombination fractions. The back-substitution may generate multiple roots because the recombination probabilities may combine in a non-linear fashion to form the conditional genotype probabilities.

In the special case of interval mapping based on an F2 sample, together with the assumptions of isolated QTL and Haldane's addition formula for recombination fractions, Whittaker *et al.* (1996) showed that a quadratic is generated by the back-substitution process. The constraint that the recombination fraction must lie in the

interval $(0, 0.5)$ rendered one root infeasible, and so a unique solution was found. The exact formula for this solution is given in the paper by Whittaker *et al.* (1996). Unfortunately, in many other cases, one is not so lucky to obtain a unique solution.

Even when a unique solution is not possible, Equation (4.46) is useful for forming a hypothesis testing strategy. This is because Equation (4.46) implies that

$$\hat{\mathbf{b}} = (\mathbf{C}_m)^+ \mathbf{W} \mathbf{C}_q (\mathbf{C}_q)^+ \boldsymbol{\mu}_q, \quad (4.51)$$

so it reveals how each marker regression coefficient captures QTL effects and it reveals where pooling and bias can occur. It initiates the process of determining whether statistically significant marker-regression coefficients are indicative of the existence of certain QTL. For an example, see Section 4.53 and Equation (4.53) below.

4.1.4 Choice of Contrasts

For t categories, a full model fits $(t - 1)$ contrasts plus the intercept, thus including t contrasts altogether. However, a reduced model that includes the intercept, fits less than $(t - 1)$ additional contrasts. Fitting a full model ensures that all main effects and all interaction effects are taken into account. If we fit a full model using different contrast matrices, the regression coefficients may differ but the same fitted means will be generated. Therefore, from a mathematical point of view, it does not strictly matter which contrast matrix is used, provided that it leads to a model matrix that is of full column rank. Nevertheless, it is beneficial to choose contrasts which yield regression coefficients that are easy to interpret.

Equation (4.22) shows that when orthogonal contrasts are used, each regression coefficient has a simple interpretation. Also, the coefficients generated by orthogonal contrasts are independent, and so the p-values from the Wald t-test for a contrast in a multiple regression and the corresponding F-test (for inclusion of that contrast)

will agree. These properties make orthogonal contrasts an appealing choice. However, orthogonal contrast matrices have the disadvantage that, by definition, their structures are sample-dependent. Therefore, the sample counts for each factor must be considered when constructing an orthogonal contrast matrix. Two algorithms for constructing an orthogonal contrast matrix are given in Appendix A of this thesis.

In QTL mapping there is a need to compare trait values amongst specific genotype groups. For certain breeding designs, this may require the use of contrasts which are not orthogonal. Tables 4.2 and 4.3 display traditional contrast coefficients that are often used in QTL mapping. A recent paper by Zeng *et al.* (2005) discusses the interpretations of several popular contrasts (coding systems) which have been proposed in the QTL mapping literature.

i	Genotype	$\mathbf{C}_{\bullet 1}$	$\mathbf{C}_{\bullet 2}$	$\mathbf{C}_{\bullet 3}$
1	$MMNN$	1	1	1
2	$MMNn$	1	0	0
3	$MmNN$	0	1	0
4	$MmNn$	0	0	0

Table 4.2: Traditional contrast coefficients to extract the main and interaction effects in a linear regression model (involving two loci M and N) for a B1 backcross sample. Note that additive and dominance effects cannot be separated because the effects of (MM vs mm) and (NN vs nn) are not estimable.

i	Genotype	$C_{\bullet 1}$	$C_{\bullet 2}$	$C_{\bullet 3}$	$C_{\bullet 4}$	$C_{\bullet 5}$	$C_{\bullet 6}$	$C_{\bullet 7}$	$C_{\bullet 8}$
1	$MMNN$	1	1	1	1	1	1	1	1
2	$MMNn$	1	1	0	-1	0	-1	0	-1
3	$MMnn$	1	1	-1	1	-1	1	-1	1
4	$MmNN$	0	-1	1	1	0	0	-1	-1
5	$MmNn$	0	-1	0	-1	0	0	0	1
6	$Mmnn$	0	-1	-1	1	0	0	1	-1
7	$mmNN$	-1	1	1	1	-1	-1	1	1
8	$mmNn$	-1	1	0	-1	0	1	0	-1
9	$mmnn$	-1	1	-1	1	1	-1	-1	1

Table 4.3: Traditional contrast coefficients to extract the additive, dominance and interaction effects in a linear regression model (involving two loci M and N) for a F2 sample.

4.2 An example based on single-marker regression

The main discussion in this section is focused on the case in which the reference population is an F2 line. Contrasts are used to estimate the dominance and additive genotypic effects of a putative quantitative trait locus in a single-marker and single-QTL model.

Consider a single marker M linked at unknown recombination fraction r to a trait locus Q and an F2 line formed from crossing MQ/MQ and mq/mq inbred lines and inbreeding the resulting F1 line. Assume that the marker alleles M and m are codominant, so that there are three observable marker genotypes in the F2 population. The three marker genotypes are MM , Mm and mm and we denote them by $i = 1, 2$, and 3 respectively. There are also three QTL genotypes, QQ , Qq and qq and we denote them by $k = 1, 2$, and 3 respectively.

Following the definitions of the additive and dominance genotypic effects given in equations (2.15) and (2.16), it is clear that the contrasts needed to estimate a the additive effect a (where $a = a_{MM}$) and the dominance effect d (where $d = d_{MM}$) are given, respectively, by the first and second columns of the matrix \mathbf{C} below.

$$\mathbf{C} = \begin{pmatrix} 1 & 1 \\ 0 & -1 \\ -1 & 1 \end{pmatrix} \text{ and } \mathbf{C}_m = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 1 & -1 & 1 \end{pmatrix}$$

If the individuals are ordered in the data so that the first n_1 are of marker genotype MM , the next n_2 of Mm and the remaining n_3 of mm then

$$\mathbf{M} = \begin{pmatrix} \mathbf{1}_{n_1} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1}_{n_2} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{1}_{n_3} \end{pmatrix}.$$

Whatever the ordering

$$\mathbf{M}^T \mathbf{M} = \begin{pmatrix} n_1 & 0 & 0 \\ 0 & n_2 & 0 \\ 0 & 0 & n_3 \end{pmatrix} \text{ and } \mathbf{1}_n^T \mathbf{M} = (n_1, n_2, n_3).$$

The i^{th} row of the model matrix, $\mathbf{X} = [\mathbf{1}_n \quad \mathbf{MC}]$, is given by

$$\mathbf{X}_{i\bullet} = \begin{cases} (1, 1, 1) & \text{if individual } i \text{ has genotype } MM, \\ (1, 0, -1) & \text{if individual } i \text{ has genotype } Mm, \\ (1, -1, 1) & \text{if individual } i \text{ has genotype } mm. \end{cases}$$

$$\begin{aligned} \mathbf{X}^T \mathbf{X} &= \begin{bmatrix} \mathbf{1}_n^T \\ \mathbf{C}^T \mathbf{M}^T \end{bmatrix} [\mathbf{1}_n \quad \mathbf{MC}] \\ &= \begin{pmatrix} n_1 + n_2 + n_3 & n_1 - n_3 & n_1 - n_2 + n_3 \\ n_1 - n_3 & n_1 + n_3 & n_1 - n_3 \\ n_1 - n_2 + n_3 & n_1 - n_3 & n_1 + n_2 + n_3 \end{pmatrix} \end{aligned}$$

The matrix $(\mathbf{X}^T \mathbf{X})^{-1}$ is equal to

$$\frac{1}{16n_1n_2n_3} \begin{pmatrix} n_1n_2 + n_2n_3 + 4n_1n_3 & 2n_2n_3 - 2n_1n_2 & n_1n_2 + n_2n_3 - 4n_1n_3 \\ 2n_2n_3 - 2n_1n_2 & 4n_1n_2 + 4n_2n_3 & 2n_2n_3 - 2n_1n_2 \\ n_1n_2 + n_2n_3 - 4n_1n_3 & 2n_2n_3 - 2n_1n_2 & n_1n_2 + n_2n_3 + 4n_1n_3 \end{pmatrix}$$

and

$$\mathbf{X}^T \mathbf{y} = \begin{pmatrix} n_1\bar{y}_1 + n_2\bar{y}_2 + n_1\bar{y}_3 \\ n_1\bar{y}_1 - n_3\bar{y}_3 \\ n_1\bar{y}_1 - n_2\bar{y}_2 + n_3\bar{y}_3 \end{pmatrix},$$

which implies that

$$\hat{\mathbf{b}} = \begin{pmatrix} \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(\bar{y}_1 + 2\bar{y}_2 + \bar{y}_3) \\ \frac{1}{2}(\bar{y}_1 - \bar{y}_3) \\ \frac{1}{4}(\bar{y}_1 - 2\bar{y}_2 + \bar{y}_3) \end{pmatrix} = \begin{pmatrix} \frac{1}{4}(\bar{y}_{MM} + 2\bar{y}_{Mm} + \bar{y}_{mm}) \\ \frac{1}{2}(\bar{y}_{MM} - \bar{y}_{mm}) \\ \frac{1}{4}(\bar{y}_{MM} - 2\bar{y}_{Mm} + \bar{y}_{mm}) \end{pmatrix}.$$

Its expected value is therefore

$$E(\hat{\mathbf{b}}) = \begin{pmatrix} \frac{1}{4}(\mu_{MM} + 2\mu_{Mm} + \mu_{mm}) \\ \frac{1}{2}(\mu_{MM} - \mu_{mm}) \\ \frac{1}{4}(\mu_{MM} - 2\mu_{Mm} + \mu_{mm}) \end{pmatrix} = \begin{pmatrix} \mu \\ a \\ d \end{pmatrix}$$

and its covariance matrix is approximately equal to $\hat{\sigma}^2(\mathbf{X}^T\mathbf{X})^{-1}$.

The overall genotypic effects are estimated by

$$\hat{\mathbf{u}}_1 = \mathbf{C}\hat{\mathbf{b}}_1 = \begin{pmatrix} \hat{a} + \hat{d} \\ -\hat{d} \\ -\hat{a} + \hat{d} \end{pmatrix},$$

where the first element estimates the effect of genotype MM , the second estimates the effect of Mm and the last estimates the effect of mm .

Standard regression produces two pieces of information, estimates for the trait means within each marker-genotype class, along with their standard errors. Regression provides estimates for marker effects and tests based on marker effects. In order to interpret these in terms of QTL effects, we let $\mathbf{C}_q = \mathbf{C}_m$, and $\hat{\mathbf{b}}_q = (\hat{\mu}, \hat{a}_{QQ}, \hat{d}_{QQ})^T$. Then, we divide the joint probabilities given in Figure 2.1(b) by the relevant marker-genotype probabilities to obtain the expression for \mathbf{W} given in Equation (4.52) below.

$$\mathbf{W} = \begin{pmatrix} (1-r)^2 & 2r(1-r) & r^2 \\ (1-r)^2 & (1-r)^2 + r^2 & (1-r)^2 \\ r^2 & 2r(1-r) & (1-r)^2 \end{pmatrix} \quad (4.52)$$

After substituting \mathbf{W} , \mathbf{C}_m , \mathbf{C}_q and $\hat{\mathbf{b}}_q$ (for \mathbf{b}_q) into Equation (4.46), we obtain the relationship

$$\hat{\mathbf{b}} = \begin{pmatrix} \hat{b}_0 \\ \hat{b}_1 \\ \hat{b}_2 \end{pmatrix} = \begin{pmatrix} \hat{\mu} \\ (1-2r)\hat{a}_{QQ} \\ (1-2r)^2\hat{d}_{QQ} \end{pmatrix}. \quad (4.53)$$

From the expected value of $\hat{\mathbf{b}}$, we see that while \hat{b}_1 and \hat{b}_2 are unbiased estimates of the marker additive and dominance effects respectively, they are downwardly biased

estimates of the corresponding QTL genotypic effects. The estimates of QTL additive and dominance effects are confounded by r , the recombination fraction, because the system of equations specified by expression (4.53) reduces to a system of two equations in three unknowns. Likewise, the estimate of r is confounded by genetic effects.

The hypothesis test for the regression coefficients tests whether $b_1 = b_2 = 0$. The existence of a linked QTL is indicated by a recombination fraction that is significantly less than 0.5 in value. Therefore, testing the hypothesis $b_1 = b_2 = 0$ is equivalent to testing that the putative QTL has no significant genetic effects on the trait or that the QTL is unlinked to the marker.

$$H_0 : b_1 = b_2 = 0 \iff H_0 : (a_{QQ} = 0 \text{ and } d_{QQ} = 0) \text{ or } r = 0.5. \quad (4.54)$$

This hypothesis test may be implemented using the F-test for regression

$$F = \frac{MS_{\text{reg}}}{MS_{\text{error}}} = \frac{(SS_{\text{total}} - SS_{\text{error}})/2}{SS_{\text{error}}/(n-3)} \sim F_{2, n-3}. \quad (4.55)$$

The null hypothesis is rejected at the α significance level if the observed value of F is greater than the $(1 - \alpha)$ -quantile of the $F_{2, n-3}$ distribution.

Simultaneous $(1 - \alpha)100\%$ confidence intervals for the b_i may be constructed using the equicorrelated multivariate-t distribution. In the single marker F2 case, we assume a bivariate t distribution to obtain confidence intervals for b_i ($i = 1, 2$) as given in Equation (4.56).

$$b_i = \hat{b}_i \pm t_{n-s, \rho_{12}(\alpha)} \sqrt{\text{var}(\hat{b}_i - \hat{b}_j)} \quad (4.56)$$

$$\text{where } \text{var}(\hat{b}_i - \hat{b}_j) = \text{var}(\hat{b}_i) + \text{var}(\hat{b}_j) - 2\text{cov}(\hat{b}_i, \hat{b}_j)$$

$$\text{var}(\hat{b}_i) \approx (\mathbf{X}^T \mathbf{X})_{ii}^{-1} SS_{\text{error}} / (n - s)$$

$$\text{cov}(\hat{b}_i, \hat{b}_j) \approx (\mathbf{X}^T \mathbf{X})_{ij}^{-1} SS_{\text{error}} / (n - s)$$

$$\rho_{12} = \text{cov}(\hat{b}_1, \hat{b}_2) / \sqrt{\text{var}(\hat{b}_1) \text{var}(\hat{b}_2)}$$

The F-test based on Equation (4.55) represents a single, joint test for significant marker effects. An alternative approach to testing the hypotheses $b_1 = 0$ and $b_2 = 0$ is to carry out multiple testing using separate Students t-tests. In a multiple testing situation, it may be necessary to make adjustments to the significance level of each test in order to ensure that the overall significance level of the combined tests is not greater than the nominal significance level of α . Common adjustments for multiple testing include making Bonferroni corrections, controlling the false discovery rate or controlling the family-wise error rate.

Under the simple Bonferroni method, the significance level for each hypothesis test is taken to be α/m where m is the number of hypotheses being tested. In a multiple testing situation with correlated hypotheses, this Bonferroni correction may produce results which are too conservative. Therefore, it is often more desirable to control the false discovery rate or the family-wise error rate.

The false discovery rate (FDR) is the expected proportion of erroneously rejected hypotheses among the list of all rejected null hypotheses (Benjamini and Hochberg, 1995). By the FDR method, a significance level for each hypothesis test is chosen under the constraint that the FDR does not exceed α .

The family-wise error rate (FWER) is the probability of having at least one falsely significant test-result within the set of hypotheses being tested (Hochberg and Tamhane, 1987). The control of the FWER is important when a conclusion from the individual null hypotheses are related (even though the different test statistics may be statistically independent). By the FWER method, a significance level for each hypothesis test is chosen under the constraint that the FWER (for each family of hypotheses) does not exceed α . In the above F2 example, rejection of either $b_1 = 0$ or $b_2 = 0$ will lead to conclusion that there is a QTL. The two hypotheses may thus be regarded as single family of hypotheses. Therefore, the use of family-wise error rates is an appropriate approach to this work.

The F2 example, presented in this chapter, is based on single-marker analysis. This single-marker methodology uses hypothesis tests based on contrasts of single-marker means as a QTL-detection strategy. Its main disadvantage is the lack of independence between the test for a linked QTL and tests for non-zero QTL effects (see Equations (4.53) and (4.54)). Lynch and Walsh (1997) describe the problem succinctly:

“A small difference between marker-homozygote means is compatible with either a tightly linked QTL of small effect or a loosely linked QTL of large effect”.

Consequently, even if the test is significant, the location of the putative QTL cannot be precisely determined from a single-marker model.

Regression on several markers has been shown to be more effective for determining QTL location than regression on one marker. For example, interval mapping by regression of a trait on two markers tends to be more powerful than single marker regression (Paterson *et al.*, 1988; Lander and Botstein, 1989; Haley and Knott, 1992; Whittaker *et al.*, 1996). The regression also can be extended to include other observed, non-genetic, explanatory variables that are thought to affect the trait.

Variance components regression models have also been used in QTL mapping. For example, Piepho (2000) proposed a mixed effects regression model to estimate QTL effects across multiple environments.

The regression can readily be adapted to a generalized linear model for binary or other categorical traits through the use of logit or probit link functions (see, for example, Hackett and Weller, 1995; Visscher *et al.*, 1996a).

Chapter 5

A Robust Interval Mapping Procedure

In this chapter, a new model for interval mapping is proposed and explored. The proposed model explicitly fits three QTL, one in the interval of interest and one on either side of it, while using marker-cofactors to control for the presence of QTL located further away. It estimates QTL position and effects by conditioning on the genotypes at four adjacent markers. These four markers define a central interval (which we will refer to as the testing interval) and its two adjacent intervals. For convenience, the proposed model (called Robust Interval Mapping Version 1.0) will be referred to by the acronym RIM1.

Composite interval mapping (CIM) is particularly susceptible to ghosting (false detections) in a situation where a QTL exists in an adjacent interval but the testing interval does not contain a QTL. The Likelihood ratio test (LRT) statistic with chi-square distribution having one degree of freedom is the null distribution generally used for null hypothesis in CIM. This null distribution is only suitable for situations where there is no QTL in any of the three intervals. It is often such a poor representation of the actual null situation that it leads to likelihood ratio tests having rates of false

positives that far exceed their nominal significance levels. Applying CIM (with LRT) to simulated data demonstrates that although the LRT performs well in isolated intervals, false positive rates as high as 100% are possible when testing non-isolated intervals. This means that while CIM (with LRT) can narrow the location of a detected QTL to the region covered by the three intervals, it cannot narrow the QTL location to the central testing interval.

The model RIM1 reduces ghosting by providing a simple multiple-QTL system that is flexible enough to model several possible null and alternative hypotheses. It also capitalizes on the strength of composite interval mapping to maintain low dimensionality in the QTL search by using marker cofactors to control the genetic background. As such, RIM1 may be viewed as an extension of CIM.

Section One of this chapter outlines the assumptions of the mixture model and details how the breeding design determines both the format of the mixing proportions and the types of genotypic effects that are estimable from the model. Section Two describes maximization of the mixture likelihood. It also tackles the onerous problem of obtaining standard errors for maximum likelihood estimators of parameters in Gaussian mixtures by giving information matrix formulae that are practical to use.

Explicit mathematical detail is given in order to show how the overall model structure can be decomposed into separate matrix systems which can then be individually modified (to support extensions) without affecting the overall format of the model. This decomposition of the overall model structure also pays dividends in simplifying the calculations of the observed and Fisher information matrices. In this chapter, the RIM1 model is presented in general terms, suitable for application to any line-cross design, but examples are only given for the B1 backcross. See Chapter 8 for extensions and for examples of applying RIM1 to the F2.

5.1 The Model Specification for RIM1

Consider four linked marker loci of known locations, denoted by K , M , N and O respectively, where the alphabetical order also indicates the marker order with K being the leftmost marker (Figure 5.1). Denote the recombination fraction between each pair of adjacent marker loci by r_{KM}, r_{MN}, r_{NO} respectively. Assume that the markers are so closely spaced that there is not likely to be more than one QTL between them. Consider also three putative QTL loci denoted by L , Q and R . Suppose that the loci are in the order $K-L-M-Q-N-R-O$, with the recombination fractions between adjacent loci given by $r_{KL}, r_{LM}, r_{MQ}, r_{QN}, r_{NR}$ and r_{RO} respectively. The resulting genetic map is shown in Figure 5.1. Note that three ordered loci, $A-B-C$ have three distances AB , BC and AC , but given any two distances and the appropriate mapping function, the third may be derived.

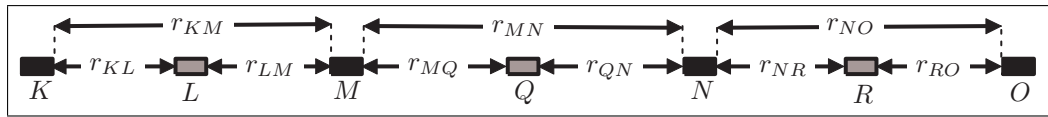


Figure 5.1: Genetic map used for modelling.

5.1.1 Genotypic content of the backcross population at the loci under study

We assume that the loci are bi-allelic and use the corresponding upper and lower case letters to denote the possible genes in the system. For example,

alleles at locus M : M, m ;

alleles at locus N : N, n ;

alleles at locus Q : Q, q .

Here we use uppercase letters to denote alleles that are present in the P1 parental genotypes and lowercase letters to denote genotypes that are present in the P2 line. Hence the P1 individuals all have genotype

$$KLMQNRO//KLMQNRO,$$

the P2 individuals all have genotype

$$klmqnro//klmqnro,$$

and the F1 individuals all have genotype

$$KLMQNRO//klmqnro.$$

The B1 backcross is formed by randomly mating P1 and F1 individuals. Therefore the possible marker genotypes in our backcross population at the pair of loci M and N (ignoring phase) are $MMNN$, $MMNn$, $MmNN$, and $MmNn$. The four marker loci, K , M , N and O taken together, yield 16 marker genotypes. There are eight possible QTL genotypes. This leads to 128 distinct genotypes at the seven loci. Table 5.1 shows the possible QTL genotypes and the labelling scheme that we use throughout this discussion. An index k ($k = 1, \dots, 8$) is used to label the QTL genotypes.

Table 5.1: QTL genotypes and their indices in a B₁-backcross model with loci in the order L - M - Q - N - R .

k	QTL genotype	k	QTL genotype
1	$LLQRRR$	5	$LlQRRR$
2	$LLQQRr$	6	$LlQQRr$
3	$LLQqRR$	7	$LlQqRR$
4	$LLQqRr$	8	$LlQqRr$

Let i index the marker genotypes where $i = 1, 2, \dots, 16$. For flexibility, when applying these methods to other designs, we also let s represent the number of marker groupings

on which we condition the QTL genotypes. Therefore, $s = 16$ in the case of a backcross design and $s = 81$ in the case of an F2 design.

5.1.2 Relating genotypic content to trait value

Let Y_{ij} be a random variable representing the trait value of individual ij , where individual ij is the j^{th} individual in marker group i . We assume that $\sum_{k=1}^t w_{ik} = 1$ and that with probability w_{ik} (for $k = 1, \dots, t$), individual ij belongs to QTL group k . We also assume that μ_{ij}^* is the cofactor effect, μ_k the QTL effect, and $\mu_{ijk} = \mu_{ij}^* + \mu_k$ is the expected trait value for a random individual having QTL genotype k , marker type i and the same cofactors as individual ij . The QTL genotypes are unobserved, therefore, we assume that with probability w_{ik} , the trait value Y_{ij} is distributed as follows:

$$Y_{ij} \sim N(\mu_{ij}^* + \mu_k, \sigma^2). \quad (5.1)$$

This leads to a Normal mixture distribution for the random trait value Y_{ij} .

Using the notation $\ddot{Y}_{ij} = Y_{ij} - \mu_{ij}^*$ we have the simpler expression

$$\ddot{Y}_{ij} \sim N(\mu_k, \sigma^2)$$

which represents the distribution of Y_{ij} (within QTL group k) after background effects have been removed. In this construction, we assume that there are no interactions between loci and that the markers are neutral. Later, in Chapter 8, we will show how to extend this model to allow for interactions between loci. Our models for the conditional trait means, $\{\mu_{ijk}\}$, and the conditional probabilities, $\{w_{ik}\}$, of the QTL genotypes given the marker genotypes are explicitly given below.

In order to specify the conditional trait means, we introduce a contrast matrix \mathbf{C} . The matrix \mathbf{C} is a device that will be used to define contrasts of the mean trait values for the QTL genotypes under study, and hence to define the complete-data

model matrix. For models having t mixing components (t QTL genotypes), \mathbf{C} will have t rows. The first column of \mathbf{C} is constrained to be a vector with all elements equal to one (to code the intercept), while the remaining columns of \mathbf{C} will depend on the contrasts of interest. The matrix \mathbf{C} is required to be of full column rank. Therefore, the maximum number of contrasts (including the intercept) in \mathbf{C} cannot be greater than the number of mixing components. Note that the matrix \mathbf{C} as presented in this chapter (and in all subsequent chapters) is conceptually equivalent to the matrix $\mathbf{C}_{\mathbf{q}}$ that was introduced in Section 4.1.3. The bold subscript in $\mathbf{C}_{\mathbf{q}}$ is henceforth dropped for convenience and for simplicity because in the current context, it is clear that we are contrasting QTL genotypic means.

Let \mathbf{b} be a vector of coefficients associated with the columns of \mathbf{C} , so that

$$E(\ddot{Y}_{ij} | \text{QTL genotype } k) = \mu_k = \mathbf{C}_{k\bullet} \mathbf{b}, \quad (5.2)$$

where $\mathbf{C}_{k\bullet}$ is k^{th} row of the matrix \mathbf{C} . We will always refer to the k^{th} row of the \mathbf{C} as $\mathbf{C}_{k\bullet}$ and its p^{th} column as $\mathbf{C}_{\bullet p}$ and we will use analogous notation when referring to the rows and columns of other matrices.

The elements of \mathbf{b} can be expressed as linear combinations of the conditional trait means. These linear combinations are derived by solving the linear system of equations defined by Equation (5.2) taken over all genotype classes (k). The expected values of the elements of \mathbf{b} may then be interpreted in terms of traditional genetic effects via any appropriate model that decomposes the conditional trait means into functions of those genetic effects.

For our backcross model, fitting no interactions between QTL, \mathbf{C} is the 8×4

contrast matrix given in Equation (5.3).

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix} \quad (5.3)$$

For our backcross example, the k^{th} row of \mathbf{C} corresponds to the k^{th} QTL genotype, where the QTL genotypes are indexed as in Table 5.1. Refer to locus L , Q and R , respectively, as the first, second and third QTL locus. For $p = 1, 2, 3$ our contrast matrix of Equation (5.3) has the property that

$$\mathbf{C}_{k(p+1)} = \begin{cases} 1, & \text{if QTL genotype } k \text{ is homozygous at the } p^{\text{th}} \text{ QTL locus} \\ 0, & \text{if QTL genotype } k \text{ is heterozygous at the } p^{\text{th}} \text{ QTL locus.} \end{cases}$$

For the backcross example, we may write

$$\mathbf{b} = (b_0, b_1, b_2, b_3)^T = (b_0, b_L, b_Q, b_R)^T, \quad (5.4)$$

and using the genetic model of Cockerham (1954) yields the interpretation of \mathbf{b} given in Equations (5.5) and (5.6) below.

Let a_{pp} be the additive effect of the homozygous-high genotype at the p^{th} QTL locus and let d_{pp} be its dominance effect. Then

$$\begin{aligned} b_0 &= E(\ddot{Y}_{ij} | k = 1) \\ &= E(\ddot{Y}_{ij} | LLQQRR) \\ &= \mu_0 - \sum_{p=1}^3 d_{pp} \end{aligned} \quad (5.5)$$

where μ_0 is the expected value of fixed effects arising from all genetic and non-genetic factors omitted from the model.

For $p = 1, 2, 3$,

$$b_p = E(\ddot{Y} | Q_p Q_p) - E(\ddot{Y} | Q_p q_p) = (a_{pp} + 2d_{pp}). \quad (5.6)$$

The remainder of this section defines a model for the mixing proportions.

The form of the conditional probabilities $\{w_{ik}\}$ of the QTL genotypes, given the marker genotypes, is determined by the recombination fractions between the marker and QTL loci, by the genetic map function and by the structure of the experimental design. Assume the classical three-locus addition formula for recombination fractions:

$$r_{MN} = r_{MQ} + r_{QN} - 2cr_{MQ}r_{QN}, \quad (5.7)$$

where c is the coefficient of coincidence.

Assume the following notation for the probability that an F1 individual transmits a certain QTL allele to an offspring, given that he/she has transmitted a particular marker haplotype to that offspring.

Define

$$p_{L1} = P(L | KM) = (1 - r_{KL} - r_{LM} + cr_{KL}r_{LM}) / (1 - r_{KM}) \quad (5.8)$$

$$p_{L2} = P(L | Km) = (1 - cr_{KL})r_{LM} / r_{KM} \quad (5.9)$$

$$p_{Q1} = P(Q | MN) = (1 - r_{MQ} - r_{QN} + cr_{MQ}r_{QN}) / (1 - r_{MN}) \quad (5.10)$$

$$p_{Q2} = P(Q | Mn) = (1 - cr_{MQ})r_{QN} / r_{MN} \quad (5.11)$$

$$p_{R1} = P(R | NO) = (1 - r_{NR} - r_{RO} + cr_{NR}r_{RO}) / (1 - r_{NO}) \quad (5.12)$$

$$p_{R2} = P(R | No) = (1 - cr_{NR})r_{RO} / r_{NO}. \quad (5.13)$$

Then the following relationships are satisfied for all three-locus genetic map functions:

$$r_{KL} = (1 - r_{KM})(1 - p_{L1}) + r_{KM}(1 - p_{L2}) \quad (5.14)$$

$$r_{LM} = (1 - r_{KM})(1 - p_{L1}) + r_{KM}(p_{L2}) \quad (5.15)$$

$$r_{MQ} = (1 - r_{MN})(1 - p_{Q1}) + r_{MN}(1 - p_{Q2}) \quad (5.16)$$

$$r_{QN} = (1 - r_{MN})(1 - p_{Q1}) + r_{MN}(p_{Q2}) \quad (5.17)$$

$$r_{NR} = (1 - r_{NO})(1 - p_{R1}) + r_{NO}(1 - p_{R2}) \quad (5.18)$$

$$r_{RO} = (1 - r_{NO})(1 - p_{R1}) + r_{NO}(p_{R2}). \quad (5.19)$$

The fact that equations (5.14) to (5.19) hold for any three-locus map function makes it possible to relax the assumption of Haldane's mapping function within the intervals L - M , M - N and N - O . However, in order for marker cofactors to absorb background genetic effects, it is necessary to assume Haldane's mapping function outside the region covered by these three intervals.

It is difficult to completely eliminate the assumption of no interference without a multi-locus feasible mapping function that is defined in the context of seven loci. The problem of finding such a mapping function is outside the scope of this thesis. Therefore, to simplify the calculation of w_{ik} , it is also necessary to assume Haldane's map function for triples of marker loci.

For convenience, let x_K , x_M , x_N and x_O , denote the genotypes at marker loci K , M , N and O respectively. Likewise, let x_L , x_Q , x_R , denote the genotypes at quantitative trait loci L , Q , R respectively. Denote the resulting four-locus marker genotype by

$$x_K x_M x_N x_O = \text{marker genotype } i \ (i = 1, \dots, 16).$$

Likewise, denote resulting three-locus QTL genotype by

$$x_L x_Q x_R = \text{QTL genotype } k \ (k = 1, \dots, t).$$

Then, the mixing proportions may be calculated as

$$w_{ik} = P(x_L | x_K, x_M) P(x_Q | x_M, x_N) P(x_R | x_N, x_O). \quad (5.20)$$

Tables 5.2 to 5.4 provide formulae for calculating the required conditional probabilities for the backcross. Table 5.5 displays the conditional probabilities w_{ik} for the Backcross example.

Table 5.2: Calculation of $P(x_L|x_K, x_M)$ for the B1 Backcross

x_K, x_M	$P(x_L = LL x_K, x_M)$	$P(x_L = Ll x_K, x_M)$
$KKMM$	p_{L1}	$1 - p_{L1}$
$KKMm$	p_{L2}	$1 - p_{L2}$
$KkMM$	$1 - p_{L2}$	p_{L2}
$KkMm$	$1 - p_{L1}$	p_{L1}

Table 5.3: Calculation of $P(x_Q|x_M, x_N)$ for the B1 Backcross

x_M, x_N	$P(x_Q = QQ x_M, x_N)$	$P(x_Q = Qq x_M, x_N)$
$MMNN$	p_{Q1}	$1 - p_{Q1}$
$MMNn$	p_{Q2}	$1 - p_{Q2}$
$MmNN$	$1 - p_{Q2}$	p_{Q2}
$MmNn$	$1 - p_{Q1}$	p_{Q1}

Table 5.4: Calculation of $P(x_R|x_N, x_O)$ for the B1 Backcross

x_N, x_O	$P(x_R = RR x_M, x_N)$	$P(x_Q = Rr x_M, x_N)$
$NNOO$	p_{R1}	$1 - p_{R1}$
$NNOo$	p_{R2}	$1 - p_{R2}$
$NnOO$	$1 - p_{R2}$	p_{R2}
$NnOo$	$1 - p_{R1}$	p_{R1}

Table 5.5: Conditional genotype probabilities, w_{ik} , for the B1 Backcross

i	Marker genotype	QTL genotypes and conditional probabilities, $w_{ik} = P(\text{QTL genotype } k \text{marker genotype } i)$.			
		$k = 1. \quad LLQQRR$ w_{i1}	$k = 2. \quad LLQQRr$ w_{i2}	$k = 3. \quad LLQqRR$ w_{i3}	$k = 4. \quad LLQqRr$ w_{i4}
1.	$KKMMNNOO$	$p_{L1}p_{Q1}p_{R1}$	$p_{L1}p_{Q1}(1-p_{R1})$	$p_{L1}(1-p_{Q1})p_{R1}$	$p_{L1}(1-p_{Q1})(1-p_{R1})$
2.	$KKMMNNOo$	$p_{L1}p_{Q1}p_{R2}$	$p_{L1}p_{Q1}(1-p_{R2})$	$p_{L1}(1-p_{Q1})p_{R2}$	$p_{L1}(1-p_{Q1})(1-p_{R2})$
3.	$KKMMNnOO$	$p_{L1}p_{Q2}(1-p_{R2})$	$p_{L1}p_{Q2}p_{R2}$	$p_{L1}(1-p_{Q2})(1-p_{R2})$	$p_{L1}(1-p_{Q2})p_{R2}$
4.	$KKMMNnoo$	$p_{L1}p_{Q2}(1-p_{R1})$	$p_{L1}p_{Q2}p_{R1}$	$p_{L1}(1-p_{Q2})(1-p_{R1})$	$p_{L1}(1-p_{Q2})p_{R1}$
5.	$KKMmNNOO$	$p_{L2}(1-p_{Q2})p_{R1}$	$p_{L2}(1-p_{Q2})(1-p_{R1})$	$p_{L2}p_{Q2}p_{R1}$	$p_{L2}p_{Q2}(1-p_{R1})$
6.	$KKMmNNOo$	$p_{L2}(1-p_{Q2})p_{R2}$	$p_{L2}(1-p_{Q2})(1-p_{R2})$	$p_{L2}p_{Q2}p_{R2}$	$p_{L2}p_{Q2}(1-p_{R2})$
7.	$KKMmNnOO$	$p_{L2}(1-p_{Q1})(1-p_{R2})$	$p_{L2}(1-p_{Q1})p_{R2}$	$p_{L2}p_{Q1}(1-p_{R2})$	$p_{L2}p_{Q1}p_{R2}$
8.	$KKMmNnoo$	$p_{L2}(1-p_{Q1})(1-p_{R1})$	$p_{L2}(1-p_{Q1})p_{R1}$	$p_{L2}p_{Q1}(1-p_{R1})$	$p_{L2}p_{Q1}p_{R1}$
9.	$KkMMNNOO$	$(1-p_{L2})p_{Q1}p_{R1}$	$(1-p_{L2})p_{Q1}(1-p_{R1})$	$(1-p_{L2})(1-p_{Q1})p_{R1}$	$(1-p_{L2})(1-p_{Q1})(1-p_{R1})$
10.	$KkMMNNOo$	$(1-p_{L2})p_{Q1}p_{R2}$	$(1-p_{L2})p_{Q1}(1-p_{R2})$	$(1-p_{L2})(1-p_{Q1})p_{R2}$	$(1-p_{L2})(1-p_{Q1})(1-p_{R2})$
11.	$KkMMNnOO$	$(1-p_{L2})p_{Q2}(1-p_{R2})$	$(1-p_{L2})p_{Q2}p_{R2}$	$(1-p_{L2})(1-p_{Q2})(1-p_{R2})$	$(1-p_{L2})(1-p_{Q2})p_{R2}$
12.	$KkMMNnoo$	$(1-p_{L2})p_{Q2}(1-p_{R1})$	$(1-p_{L2})p_{Q2}p_{R1}$	$(1-p_{L2})(1-p_{Q2})(1-p_{R1})$	$(1-p_{L2})(1-p_{Q2})p_{R1}$
13.	$KkMmNNOO$	$(1-p_{L1})(1-p_{Q2})p_{R1}$	$(1-p_{L1})(1-p_{Q2})(1-p_{R1})$	$(1-p_{L1})p_{Q2}p_{R1}$	$(1-p_{L1})p_{Q2}(1-p_{R1})$
14.	$KkMmNNOo$	$(1-p_{L1})(1-p_{Q2})p_{R2}$	$(1-p_{L1})(1-p_{Q2})(1-p_{R2})$	$(1-p_{L1})p_{Q2}p_{R2}$	$(1-p_{L1})p_{Q2}(1-p_{R2})$
15.	$KkMmNnOO$	$(1-p_{L1})(1-p_{Q1})(1-p_{R2})$	$(1-p_{L1})(1-p_{Q1})p_{R2}$	$(1-p_{L1})p_{Q1}(1-p_{R2})$	$(1-p_{L1})p_{Q1}p_{R2}$
16.	$KkMmNnoo$	$(1-p_{L1})(1-p_{Q1})(1-p_{R1})$	$(1-p_{L1})(1-p_{Q1})p_{R1}$	$(1-p_{L1})p_{Q1}(1-p_{R1})$	$(1-p_{L1})p_{Q1}p_{R1}$
		$k = 5. \quad LlQQRR$ w_{i5}	$k = 6. \quad LlQQRr$ w_{i6}	$k = 7. \quad LlQqRR$ w_{i7}	$k = 8. \quad LlQqRr$ w_{i8}
1.	Marker genotype				
1.	$KKMMNNOO$	$(1-p_{L1})p_{Q1}p_{R1}$	$(1-p_{L1})p_{Q1}(1-p_{R1})$	$(1-p_{L1})(1-p_{Q1})p_{R1}$	$(1-p_{L1})(1-p_{Q1})(1-p_{R1})$
2.	$KKMMNNOo$	$(1-p_{L1})p_{Q1}p_{R2}$	$(1-p_{L1})p_{Q1}(1-p_{R2})$	$(1-p_{L1})(1-p_{Q1})p_{R2}$	$(1-p_{L1})(1-p_{Q1})(1-p_{R2})$
3.	$KKMMNnOO$	$(1-p_{L1})p_{Q2}(1-p_{R2})$	$(1-p_{L1})p_{Q2}p_{R2}$	$(1-p_{L1})(1-p_{Q2})(1-p_{R2})$	$(1-p_{L1})(1-p_{Q2})p_{R2}$
4.	$KKMMNnoo$	$(1-p_{L1})p_{Q2}(1-p_{R1})$	$(1-p_{L1})p_{Q2}p_{R1}$	$(1-p_{L1})(1-p_{Q2})(1-p_{R1})$	$(1-p_{L1})(1-p_{Q2})p_{R1}$
5.	$KKMmNNOO$	$(1-p_{L2})(1-p_{Q2})p_{R1}$	$(1-p_{L2})(1-p_{Q2})(1-p_{R1})$	$(1-p_{L2})p_{Q2}p_{R1}$	$(1-p_{L2})p_{Q2}(1-p_{R1})$
6.	$KKMmNNOo$	$(1-p_{L2})(1-p_{Q2})p_{R2}$	$(1-p_{L2})(1-p_{Q2})(1-p_{R2})$	$(1-p_{L2})p_{Q2}p_{R2}$	$(1-p_{L2})p_{Q2}(1-p_{R2})$
7.	$KKMmNnOO$	$(1-p_{L2})(1-p_{Q1})(1-p_{R2})$	$(1-p_{L2})(1-p_{Q1})p_{R2}$	$(1-p_{L2})p_{Q1}(1-p_{R2})$	$(1-p_{L2})p_{Q1}p_{R2}$
8.	$KKMmNnoo$	$(1-p_{L2})(1-p_{Q1})(1-p_{R1})$	$(1-p_{L2})(1-p_{Q1})p_{R1}$	$(1-p_{L2})p_{Q1}(1-p_{R1})$	$(1-p_{L2})p_{Q1}p_{R1}$
9.	$KkMMNNOO$	$p_{L2}p_{Q1}p_{R1}$	$p_{L2}p_{Q1}(1-p_{R1})$	$p_{L2}(1-p_{Q1})p_{R1}$	$p_{L2}(1-p_{Q1})(1-p_{R1})$
10.	$KkMMNNOo$	$p_{L2}p_{Q1}p_{R2}$	$p_{L2}p_{Q1}(1-p_{R2})$	$p_{L2}(1-p_{Q1})p_{R2}$	$p_{L2}(1-p_{Q1})(1-p_{R2})$
11.	$KkMMNnOO$	$p_{L2}p_{Q2}(1-p_{R2})$	$p_{L2}p_{Q2}p_{R2}$	$p_{L2}(1-p_{Q2})(1-p_{R2})$	$p_{L2}(1-p_{Q2})p_{R2}$
12.	$KkMMNnoo$	$p_{L2}p_{Q2}(1-p_{R1})$	$p_{L2}p_{Q2}p_{R1}$	$p_{L2}(1-p_{Q2})(1-p_{R1})$	$p_{L2}(1-p_{Q2})p_{R1}$
13.	$KkMmNNOO$	$p_{L1}(1-p_{Q2})p_{R1}$	$p_{L1}(1-p_{Q2})(1-p_{R1})$	$p_{L1}p_{Q2}p_{R1}$	$p_{L1}p_{Q2}(1-p_{R1})$
14.	$KkMmNNOo$	$p_{L1}(1-p_{Q2})p_{R2}$	$p_{L1}(1-p_{Q2})(1-p_{R2})$	$p_{L1}p_{Q2}p_{R2}$	$p_{L1}p_{Q2}(1-p_{R2})$
15.	$KkMmNnOO$	$p_{L1}(1-p_{Q1})(1-p_{R2})$	$p_{L1}(1-p_{Q1})p_{R2}$	$p_{L1}p_{Q1}(1-p_{R2})$	$p_{L1}p_{Q1}p_{R2}$
16.	$KkMmNnoo$	$p_{L1}(1-p_{Q1})(1-p_{R1})$	$p_{L1}(1-p_{Q1})p_{R1}$	$p_{L1}p_{Q1}(1-p_{R1})$	$p_{L1}p_{Q1}p_{R1}$

Next, we examine the properties p_{Q1} and p_{Q2} in order to determine what constraints to place on these mixing parameters.

Properties of p_{Q1} :

- $p_{Q1} = 1 - cr_{MQ}r_{QN}/(1 - r_{MN})$.
- $\max(p_{Q1}) = 1$ and occurs if $r_{MQ} = 0$ or $r_{MQ} = r_{MN}$ (since $r_{QN} = 0$ when $r_{MQ} = r_{MN}$).
- For a fixed r_{MN} and fixed c , the value of p_{Q1} is minimized when the product $cr_{MQ}r_{QN}$ is maximized. Therefore, p_{Q1} is minimized when

$$r_{MQ} = \frac{1}{2c}(1 - \sqrt{1 - 2cr_{MN}}).$$

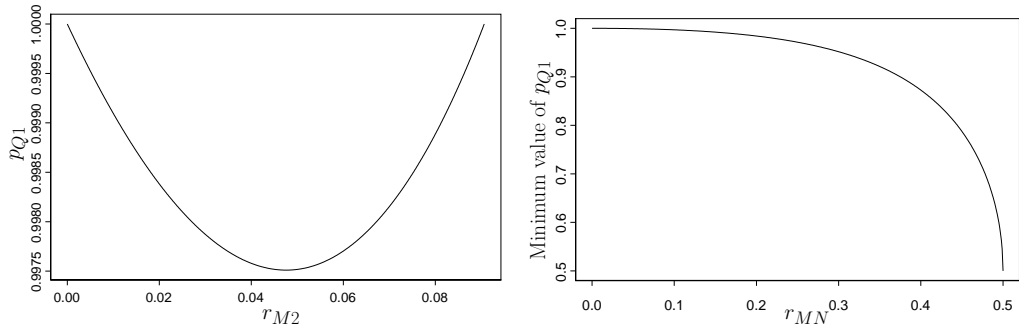


Figure 5.2: A plot of p_{Q1} versus r_{MQ} for markers 10 centiMorgans apart ($r_{MN} = 0.0906$), and a plot of $\min(p_{Q1})$ versus r_{MN} .

If Haldane's addition formula holds, then $c = 1$ and p_{Q1} is minimized when the QTL is exactly in the middle of the interval so that

$$r_{MQ} = r_{QN} = \frac{1}{2}(1 - \sqrt{1 - 2r_{MN}}) \text{ and}$$

$$\min(p_{Q1}) = 1 - \frac{1}{4}(1 - \sqrt{1 - 2r_{MN}})^2/(1 - r_{MN}).$$

As r_{MN} increases $\min(p_{Q1})$ decreases and $0 \leq r_{MN} \leq 0.5$. Therefore, $\min(p_{Q1}) \geq 0.5$. See Figure 5.2 for an illustration.

Properties of p_{Q2} :

- p_{Q2} is monotonic decreasing (as a function of r_{MQ}) on $[0, r_{MN}]$. See Figure 5.3 for an example.

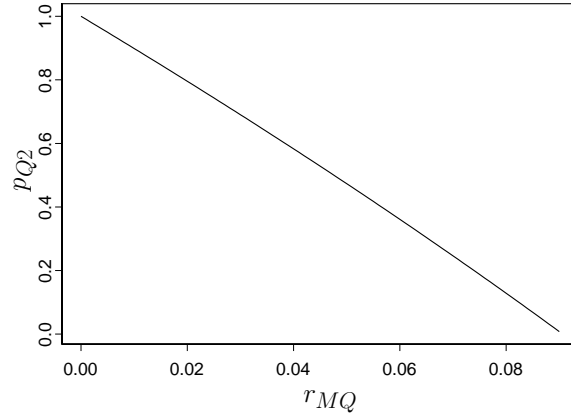


Figure 5.3: Plot of p_{Q2} versus r_{MQ} for markers 10 centiMorgans apart ($r_{MN} = 0.0906$).

If we choose not to assume any specific map function within the three intervals, then the vector of mixing parameters is $\phi = (p_{L1}, p_{L2}, p_{Q1}, p_{Q2}, p_{R1}, p_{R2})^T$. If Haldane's map function is used, then the relationships given in Equations (5.21) to (5.23) hold, and the vector of mixing parameters reduces to $\phi = (p_{L2}, p_{Q2}, p_{R2})^T$ and then

$$p_{L1} = \frac{1}{2} + \frac{\sqrt{1 - 2r_{KM} + r_{KM}^2(1 - 2p_{L2})^2}}{2(1 - r_{KM})}, \quad (5.21)$$

$$p_{Q1} = \frac{1}{2} + \frac{\sqrt{1 - 2r_{MN} + r_{MN}^2(1 - 2p_{Q2})^2}}{2(1 - r_{MN})}, \quad (5.22)$$

$$p_{R1} = \frac{1}{2} + \frac{\sqrt{1 - 2r_{NO} + r_{NO}^2(1 - 2p_{R2})^2}}{2(1 - r_{NO})}. \quad (5.23)$$

By construction, for any fixed marker category i , the conditional genotype probabilities $\{w_{ik} : k = 1, \dots, t\}$ sum to one. Our model also requires that each w_{ik} is strictly

greater than zero and strictly less than one. This imposes the following constraints on the mixing parameters:

$$0 < p_{L1}, p_{Q1}, p_{R1} < 1$$

$$0 < p_{L2}, p_{Q2}, p_{R2} < 1$$

The impact of these constraints is to place the first and third QTL loci (L and R) strictly exterior to the testing interval ($M-N$) and to place the second QTL locus (Q) strictly interior to it.

5.1.3 The model matrix and likelihood function for a sample

Let y_{ij} be the trait value of the j^{th} individual with marker genotype i for $i = 1, \dots, s$ and $j = 1, \dots, n_i$ where n_i is the number of individuals having marker genotype i . The overall sample size is given by $n = \sum_{i=1}^s n_i$, and the observed trait values are organized to form a vector \mathbf{y} where

$$\mathbf{y} = (y_{11}, \dots, y_{1n_1}, y_{21}, \dots, y_{2n_2}, \dots, y_{s1}, \dots, y_{s,n_s})^T. \quad (5.24)$$

Define $\mathbf{Z}_{(ij)\bullet}$ to be a (row) vector-valued random variable indicating the QTL-category identity of observation y_{ij} , where

$$\mathbf{Z}_{(ij)\bullet} = (z_{ij1}, z_{ij2}, \dots, z_{ijt}), \text{ for } i = 1, \dots, s \text{ and } j = 1, \dots, n_i$$

$$\text{and } z_{ijk} = \begin{cases} 1, & \text{if } y_{ij} \text{ belongs to QTL-category } k \\ 0 & \text{otherwise.} \end{cases}$$

$$\mathbf{Z}_{(ij)\bullet} \sim \text{Multinomial}(1; w_{i1}, \dots, w_{it})$$

The vector $\mathbf{Z}_{(ij)\bullet}$ indicates the overall QTL genotype of an individual. However, for our backcross example, the components of $\mathbf{Z}_{(ij)\bullet}$ may be combined to form indicators

for the genotypes at the separate QTL loci because the quantity $\mathbf{Z}_{(ij)\bullet}\mathbf{C}_{\bullet(p+1)}$ where $\mathbf{C}_{\bullet(p+1)}$ is the $(p+1)^{\text{th}}$ column of \mathbf{C} (for $p = 1, 2, 3$) has the property that

$$\mathbf{Z}_{(ij)\bullet}\mathbf{C}_{\bullet(p+1)} = \begin{cases} 1, & \text{if individual } ij \text{ is homozygous at locus } Q_p \\ 0, & \text{otherwise.} \end{cases}$$

The intercept and QTL effects are encapsulated in the vector of coefficients denoted by \mathbf{b} (see Equation (5.2)).

Zeng (1994) has shown that if recombination fractions obey Haldane's addition formula, then marker cofactors can absorb the effects of background QTL that are not included in a linear model. With real data, Haldane's map function will, at best, only approximate the true situation. Consequently, while it is true that marker cofactors can help to control for background QTL effects, marker cofactors may not completely absorb background genetic effects.

To facilitate the inclusion of background markers as extra cofactors, we introduce a matrix of cofactors \mathbf{X}_2 , and another set of coefficients, \mathbf{b}^* associated with its columns. Non-genetic factors may also be included as columns of \mathbf{X}_2 , if desired. The model for an individual trait value is:

$$y_{ij} = \mathbf{Z}_{(ij)\bullet}\mathbf{C}\mathbf{b} + [\mathbf{X}_2]_{(ij)\bullet}\mathbf{b}^* + \varepsilon_{ij} \quad (5.25)$$

where the values $\{\varepsilon_{ij} : i = 1, 2, \dots, s; j = 1, 2, \dots, n_i\}$ are observations of independent identically distributed random variables, having Normal distribution with variance equal to σ^2 and mean equal to zero.

It is convenient to separate the conventional part of the regression and those parts of the linear model which capture QTL effects. To achieve this separation, we define the centered data:

$$\ddot{y}_{ij} = y_{ij} - [\mathbf{X}_2]_{(ij)\bullet}\mathbf{b}^* = y_{ij} - \mu_{ij}^* \quad (5.26)$$

and write Equation (5.25) as

$$\ddot{y}_{ij} = \mathbf{Z}_{(ij)\bullet}\mathbf{C}\mathbf{b} + \varepsilon_{ij}. \quad (5.27)$$

Denote the $(n \times t)$ matrix of missing data by

$$\mathbf{Z} = (\mathbf{Z}_{(11)\bullet}^T, \dots, \mathbf{Z}_{(1n_1)\bullet}^T, \mathbf{Z}_{(21)\bullet}^T, \dots, \mathbf{Z}_{(2n_2)\bullet}^T, \dots, \mathbf{Z}_{(s1)\bullet}^T, \dots, \mathbf{Z}_{(sn_s)\bullet}^T)^T. \quad (5.28)$$

Then the complete-data model matrix may be written simply as

$$\mathbf{X} = [\mathbf{X}_1 \quad \mathbf{X}_2] = [\mathbf{ZC} \quad \mathbf{X}_2] \quad (5.29)$$

and the model parameters as

$$\boldsymbol{\psi} = \begin{bmatrix} \boldsymbol{\beta} \\ \sigma^2 \\ \boldsymbol{\phi} \end{bmatrix}, \text{ where } \boldsymbol{\beta} = (\mathbf{b}, \mathbf{b}^*)^T. \quad (5.30)$$

The aim is to determine QTL location and effects by estimating the components of $\boldsymbol{\psi}$, the parameter vector. We take a maximum likelihood approach to parameter estimation. The likelihood functions under consideration are presented below.

The probability density function of the trait value y_{ij} conditional on $\mathbf{Z}_{(ij)\bullet}$ is equal to

$$p(y_{ij} | \mathbf{Z}_{(ij)\bullet}; \boldsymbol{\beta}, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ \frac{(y_{ij} - \mathbf{Z}_{(ij)\bullet}\mathbf{C}\mathbf{b} - [\mathbf{X}_2]_{(ij)\bullet}\mathbf{b}^*)^2}{-2\sigma^2} \right\}$$

and the probability mass function of $\mathbf{Z}_{(ij)\bullet}$ is given by the multinomial formula

$$p(\mathbf{Z}_{(ij)\bullet}; \boldsymbol{\phi}) = \prod_{k=1}^t w_{ik}^{z_{ijk}}.$$

Therefore their joint density is equal to

$$f_c(y_{ij}, \mathbf{Z}_{(ij)\bullet}; \boldsymbol{\psi}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{ \frac{(y_{ij} - \mathbf{Z}_{(ij)\bullet}\mathbf{C}\mathbf{b})^2}{-2\sigma^2} \right\} \prod_{k=1}^t w_{ik}^{z_{ijk}}.$$

Given the complete genotype information at both QTL and marker loci, the (complete-data) likelihood function may be expressed as follows.

$$\mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) = (\sigma\sqrt{2\pi})^{-n} \prod_{i=1}^s \prod_{j=1}^{n_i} \left(\exp\left\{ \frac{(y_{ij} - \mathbf{Z}_{(ij)\bullet}\mathbf{C}\mathbf{b})^2}{-2\sigma^2} \right\} \prod_{k=1}^t w_{ik}^{z_{ijk}} \right). \quad (5.31)$$

If we had the complete data, then $\hat{\mathbf{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}$ would represent the fitted values from this model, and the estimates $\hat{\boldsymbol{\beta}}$ could be obtained by a linear regression of \mathbf{y} on \mathbf{X} . However, unlike the usual case for a general linear model, part of our design matrix \mathbf{X} is unobserved because the $\mathbf{Z}_{(ij)\bullet}$ are unknown.

Summing the joint density, $f_c(y_{ij}, \mathbf{Z}_{(ij)\bullet}; \boldsymbol{\psi})$, over the t possible values of $\mathbf{Z}_{(ij)\bullet}$, we obtain the marginal density of an individual's trait value, y_{ij} , as the mixture density given in Equation (5.32).

$$f(y_{ij}; \boldsymbol{\psi}) = \frac{1}{\sigma\sqrt{2\pi}} \sum_{k=1}^t w_{ik} \exp\left\{\frac{(\ddot{y}_{ij} - \mathbf{C}_{k\bullet}\mathbf{b})^2}{-2\sigma^2}\right\} \quad (5.32)$$

$$= \sum_{k=1}^t w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi}),$$

$$\text{where } f_{ik}(y_{ij}; \boldsymbol{\psi}) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{\frac{(\ddot{y}_{ij} - \mathbf{C}_{k\bullet}\mathbf{b})^2}{-2\sigma^2}\right\}. \quad (5.33)$$

Therefore the observed (incomplete) dataset has the following mixture likelihood:

$$\mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) = (\sigma\sqrt{2\pi})^{-n} \prod_{i=1}^s \prod_{j=1}^{n_i} \left(\sum_{k=1}^t w_{ik} \exp\left\{\frac{(\ddot{y}_{ij} - \mathbf{C}_{k\bullet}\mathbf{b})^2}{-2\sigma^2}\right\} \right). \quad (5.34)$$

The mixture likelihood above is the function from which we seek maximum likelihood estimates of model parameters $(\boldsymbol{\beta}, \sigma^2, \boldsymbol{\phi})$, in order to determine QTL locations and effects.

5.2 Maximum Likelihood Analysis

5.2.1 Maximization Procedure

We wish to maximize mixture likelihood given in Equation (5.34), which is the likelihood of the observed data. The system of equations obtained by setting the scores of this likelihood to zero does not yield an explicit solution for the maximum likelihood

estimates (MLEs) of the parameters. Consequently, this likelihood is computationally demanding and is also often unstable to maximize via the usual derivative-based methods, such as the Newton Raphson procedure. However, a useful feature of missing data models such as ours, is that the score of the observed likelihood is equal to the conditional expectation of the score of the complete-data likelihood, given the observed data (see, for example, McLachlan and Krishnan 1996, page 100).

$$\frac{\partial}{\partial \boldsymbol{\psi}} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} \left[\frac{\partial}{\partial \boldsymbol{\psi}} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) \right]$$

This well known and important result, which we prove below as Result 5.2.1, is useful because it provides simple expressions for first derivatives of the observed likelihood, and it allows us to express the maximum likelihood estimates of the parameters as functions of the expected values of the missing data (\mathbf{Z}). This is a recursive solution because the expectations of the components of \mathbf{Z} are also functions of the model parameters.

The E-step of the EM algorithm of Dempster *et al.* (1977) provides a mechanism for estimating the expected value of the missing data using initial or updated parameter estimates. The M-step finds new parameter estimators by calculating the MLE from the scores the the observed likelihood. It exploits the fact that these are simply functions of the expected values of the missing data, which were calculated in the E-Step. Below, we calculate the complete-data and observed likelihood functions and demonstrate how the EM algorithm should be applied to our model.

Let t be the number of components (QTL genotypes) in the mixture. Then the natural logarithm of the complete-data likelihood is given by

$$\begin{aligned} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) = & -n \ln \sqrt{2\pi} - n \ln \sigma + \sum_{i=1}^s \sum_{j=1}^{n_i} \sum_{k=1}^t z_{ijk} \ln w_{ik} \\ & + \sum_{i=1}^s \sum_{j=1}^{n_i} \frac{1}{-2\sigma^2} \left(y_{ij} - \mathbf{Z}_{(ij)\bullet} \mathbf{C} \mathbf{b} - [\mathbf{X}_2]_{(ij)\bullet} \mathbf{b}^* \right)^2. \end{aligned}$$

To write this in matrix form, let

$$\mathbf{w}_i(\boldsymbol{\phi}) = (w_{i1}, w_{i2}, \dots, w_{it})^T, \quad (5.35)$$

$$\mathbf{h}_i(\boldsymbol{\phi}) = (\ln w_{i1}, \ln w_{i2}, \dots, \ln w_{it})^T, \quad (5.36)$$

$$\mathbf{Z}_i = (\mathbf{Z}_{(i1)\bullet}^T, \mathbf{Z}_{(i2)\bullet}^T, \dots, \mathbf{Z}_{(in_i)\bullet}^T)^T, \quad (5.37)$$

and let $\mathbf{1}_{n_i}$ be a column vector of order n_i with each element equal to one.

The matrix \mathbf{Z}_i is the i^{th} block of the (unobserved) matrix of indicators \mathbf{Z} and

$$\mathbf{Z}_i = \boldsymbol{\Delta}_i \mathbf{Z} \quad (5.38)$$

where $\boldsymbol{\Delta}_i$ is an $n_i \times n$ matrix which is a partition of the identity matrix, \mathbf{I}_n , of order n such that (for $i = 1, 2, \dots, s$)

$$\mathbf{I}_n = \begin{bmatrix} \boldsymbol{\Delta}_1 \\ \boldsymbol{\Delta}_2 \\ \vdots \\ \boldsymbol{\Delta}_s \end{bmatrix}. \quad (5.39)$$

Denote the number of individuals belonging to both marker-group i and QTL-group k by n_{ik} where

$$n_{ik} = \mathbf{1}_{n_i}^T (\mathbf{Z}_i)_{\bullet k} = \sum_{j=1}^{n_i} z_{ijk}. \quad (5.40)$$

Denote the number of individuals QTL-group k by m_k where

$$m_k = \mathbf{1}_n^T \mathbf{Z}_{\bullet k} = \sum_{i=1}^s n_{ik}. \quad (5.41)$$

Then, from equations (5.38) to (5.41), we obtain the following identities.

$$\mathbf{Z}_i^T \mathbf{1}_{n_i} = \mathbf{Z}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i} = (n_{i1}, n_{i2}, \dots, n_{it})^T \quad (5.42)$$

$$\mathbf{1}_n^T \mathbf{Z} = (m_1, m_2, \dots, m_t) \quad (5.43)$$

Now we may write the complete-data log-likelihood in matrix form.

$$\begin{aligned} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) &= -n \ln \sqrt{2\pi} - n \ln \sigma + \sum_{i=1}^s \mathbf{h}_i^T(\phi) \mathbf{Z}_i^T \mathbf{1}_{n_i} \\ &\quad - \frac{1}{2\sigma^2} (\mathbf{y} - \mathbf{X}_1 \mathbf{b} - \mathbf{X}_2 \mathbf{b}^*)^T (\mathbf{y} - \mathbf{X}_1 \mathbf{b} - \mathbf{X}_2 \mathbf{b}^*) \end{aligned} \quad (5.44)$$

$$\begin{aligned} &= -n \ln \sqrt{2\pi} - n \ln \sigma + \sum_{i=1}^s \mathbf{h}_i^T(\phi) \mathbf{Z}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i} \\ &\quad - \frac{1}{2\sigma^2} (\ddot{\mathbf{y}} - \mathbf{ZC} \mathbf{b})^T (\ddot{\mathbf{y}} - \mathbf{ZC} \mathbf{b}), \text{ where } \ddot{\mathbf{y}} = \mathbf{y} - \mathbf{X}_2 \mathbf{b}^* \end{aligned} \quad (5.45)$$

Note that $(\mathbf{y} - \mathbf{X}_1 \mathbf{b} - \mathbf{X}_2 \mathbf{b}^*) = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$.

The EM algorithm works with a synthetic, complete-data, design matrix constructed by replacing the missing $\{z_{ijk}\}$ with their expected values conditioned on the observed data (trait values and marker genotypes). From the definition of z_{ijk} we have that

$$E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}(z_{ijk}) = E(z_{ijk} | y_{ij}; \boldsymbol{\psi}) = P(z_{ijk} = 1 | y_{ij}; \boldsymbol{\psi}).$$

Let $\tau_{ik}(y_{ij}; \boldsymbol{\psi}) = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}(z_{ijk})$, then by Bayes theorem,

$$\tau_{ik}(y_{ij}; \boldsymbol{\psi}) = \frac{P(z_{ijk} = 1)P(y_{ij} | z_{ijk} = 1)}{\sum_{k=1}^t P(z_{ijk} = 1)P(y_{ij} | z_{ijk} = 1)}. \quad (5.46)$$

Therefore, for our model,

$$\tau_{ik}(y_{ij}; \boldsymbol{\psi}) = \frac{w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi})}{f(y_{ij}; \boldsymbol{\psi})} = \frac{w_{ik} \exp\left\{\frac{(y_{ij} - [\mathbf{X}_2]_{(ij)\bullet} \mathbf{b}^* - \mathbf{C}_{k\bullet} \mathbf{b})^2}{-2\sigma^2}\right\}}{\sum_{k=1}^t w_{ik} \exp\left\{\frac{(y_{ij} - [\mathbf{X}_2]_{(ij)\bullet} \mathbf{b}^* - \mathbf{C}_{k\bullet} \mathbf{b})^2}{-2\sigma^2}\right\}}. \quad (5.47)$$

Note that $\sum_{k=1}^t \tau_{ik}(y_{ij}; \boldsymbol{\psi}) = 1$ and that $P(z_{ijk} = 1 | y_{ij}; \boldsymbol{\psi}) = w_{ik}$.

We see that the expected values of the QTL category identities are determined by the recombination fractions which determine the mixing proportions, by QTL effects, and by the effects of all extra cofactors (genetic or non-genetic) in our model.

Let

$$\tilde{\mathbf{Z}}_{(ij)\bullet} = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}(\mathbf{Z}_{(ij)\bullet}) = (\tau_{i1}(y_{ij}; \boldsymbol{\psi}), \tau_{i2}(y_{ij}; \boldsymbol{\psi}), \dots, \tau_{it}(y_{ij}; \boldsymbol{\psi})) \quad (5.48)$$

and for the i^{th} block (\mathbf{Z}_i) of the missing data matrix (\mathbf{Z}), let

$$\tilde{\mathbf{Z}}_i = E_{\mathbf{Z}|\mathbf{y};\psi}(\mathbf{Z}_i) = (\tilde{\mathbf{Z}}_{(i1)\bullet}^T, \tilde{\mathbf{Z}}_{(i2)\bullet}^T, \dots, \tilde{\mathbf{Z}}_{(in_i)\bullet}^T)^T = \mathbf{\Delta}_i \tilde{\mathbf{Z}}. \quad (5.49)$$

Also, let $\tilde{\mathbf{Z}} = E_{\mathbf{Z}|\mathbf{y};\psi}(\mathbf{Z})$. Then, representing $\tilde{\mathbf{Z}}$ as a partitioned matrix, we have

$$\tilde{\mathbf{Z}} = \begin{bmatrix} \tilde{\mathbf{Z}}_1 \\ \tilde{\mathbf{Z}}_2 \\ \vdots \\ \tilde{\mathbf{Z}}_s \end{bmatrix}. \quad (5.50)$$

The complete-data design matrix given in Equation (5.29) cannot be used directly because it depends on the unknown matrix of category identities.

Note that \mathbf{Z} is an unobservable binary indicator matrix (containing only zeros and ones) so it is estimated by the imputed matrix $\tilde{\mathbf{Z}}$ (which can contain fractions). Both matrices, \mathbf{Z} and $\tilde{\mathbf{Z}}$, have each row summing to one. The estimated complete-data design matrix is then

$$\tilde{\mathbf{X}} = [\tilde{\mathbf{Z}}\mathbf{C} \quad \mathbf{X}_2]. \quad (5.51)$$

Now,

$$\mathbf{X}^T\mathbf{X} = \begin{bmatrix} \mathbf{C}^T\mathbf{Z}^T\mathbf{Z}\mathbf{C} & \mathbf{C}^T\mathbf{Z}^T\mathbf{X}_2 \\ \mathbf{X}_2^T\mathbf{Z}\mathbf{C} & \mathbf{X}_2^T\mathbf{X}_2 \end{bmatrix}. \quad (5.52)$$

Therefore we will also need to evaluate $E_{\mathbf{Z}|\mathbf{y};\psi}(\mathbf{Z}^T\mathbf{Z})$.

The element in row k and column k' (for $k, k' = 1, \dots, t$) of the matrix $\mathbf{Z}^T\mathbf{Z}$ is given by

$$\begin{aligned} (\mathbf{Z}^T\mathbf{Z})_{kk'} &= \sum_{i=1}^s \sum_{j=1}^{n_i} z_{ijk} z_{ijk'} \\ &= \begin{cases} \sum_{i=1}^s \sum_{j=1}^{n_i} z_{ijk} = \mathbf{1}_n^T \mathbf{Z}_{\bullet k} = m_k, & \text{if } k = k' \\ 0, & \text{if } k \neq k'. \end{cases} \end{aligned}$$

Therefore, the matrix $\mathbf{Z}^T\mathbf{Z}$ is diagonal with the k^{th} diagonal element equal to the overall number of sampled individuals, m_k , having the k^{th} QTL genotype.

Now, we introduce notation for constructing a diagonal matrix from the elements of a row vector. This notation will greatly simplify calculations later. Suppose that $\mathbf{v}^T = (v_1, v_2, \dots, v_\xi)$ is a row vector of order ξ . Then let $\text{diag}(\mathbf{v}^T)$ denote the diagonal matrix (of order ξ) whose i^{th} diagonal element is given by the i^{th} element of \mathbf{v}^T . Therefore

$$\text{diag}(\mathbf{v}^T) = \begin{pmatrix} v_1 & 0 & \mathbf{0} & 0 \\ 0 & v_2 & \mathbf{0} & 0 \\ \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} \\ 0 & 0 & \mathbf{0} & v_\xi \end{pmatrix}. \quad (5.53)$$

In fact, since the k^{th} diagonal element of the diagonal matrix $\mathbf{Z}^T \mathbf{Z}$ is the k^{th} element of the row vector $\mathbf{1}_n^T \mathbf{Z}$, we write

$$\mathbf{Z}^T \mathbf{Z} = \text{diag}(\mathbf{1}_n^T \mathbf{Z}) \quad (5.54)$$

and so

$$E_{\mathbf{Z}|\mathbf{y};\psi}(\mathbf{Z}^T \mathbf{Z}) = \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}). \quad (5.55)$$

Therefore, $E_{\mathbf{Z}|\mathbf{y};\psi}(\mathbf{Z}^T \mathbf{Z})$ is a diagonal matrix with its k^{th} diagonal element equal to the expected number of sampled individuals, \tilde{m}_k , having the k^{th} QTL genotype conditioned on the observed data.

Let

$$\widetilde{(\mathbf{X}^T \mathbf{X})} = E_{\mathbf{Z}|\mathbf{y};\psi}(\mathbf{X}^T \mathbf{X}).$$

Then

$$\widetilde{(\mathbf{X}^T \mathbf{X})} = \begin{bmatrix} \mathbf{C}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \mathbf{C} & \mathbf{C}^T \tilde{\mathbf{Z}}^T \mathbf{X}_2 \\ \mathbf{X}_2^T \tilde{\mathbf{Z}} \mathbf{C} & \mathbf{X}_2^T \mathbf{X}_2 \end{bmatrix}. \quad (5.56)$$

Note that $E_{\mathbf{Z}|\mathbf{y};\psi}(\mathbf{Z}^T \mathbf{Z}) \neq \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}}$ and likewise $\widetilde{(\mathbf{X}^T \mathbf{X})} \neq \tilde{\mathbf{X}}^T \tilde{\mathbf{X}}$.

Next, denote the first and second derivatives of the observed log-likelihood as

follows:

$$\mathcal{S}(\boldsymbol{\psi}; \mathbf{y}) = \frac{\partial}{\partial \boldsymbol{\psi}} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \quad (5.57)$$

$$-\mathcal{I}(\boldsymbol{\psi}; \mathbf{y}) = \frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}). \quad (5.58)$$

For the complete-data log-likelihood, let

$$\mathcal{S}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z}) = \frac{\partial}{\partial \boldsymbol{\psi}} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) = \mathcal{U}_\psi \quad (5.59)$$

$$-\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z}) = \frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) = \mathcal{U}_{\psi\psi} \quad (5.60)$$

Now we show that the conditional expectation of the score of the complete-data likelihood, given the observed data is equal to the score of the observed likelihood.

Result 5.2.1. $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_\psi] = \mathcal{S}(\boldsymbol{\psi}; \mathbf{y})$

Proof of Result 5.2.1.

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_\psi] &= \int \left(\frac{\partial}{\partial \boldsymbol{\psi}} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) \right) \frac{\mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi})}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} d\mathbf{Z} \\ &= \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \int \left(\frac{\partial}{\partial \boldsymbol{\psi}} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) \right) \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) d\mathbf{Z} \\ &= \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \int \left(\frac{\partial}{\partial \boldsymbol{\psi}} \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) \right) d\mathbf{Z} \\ &= \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \left(\frac{\partial}{\partial \boldsymbol{\psi}} \int \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) d\mathbf{Z} \right) \\ &= \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \left(\frac{\partial}{\partial \boldsymbol{\psi}} \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \right) \\ &= \frac{\partial}{\partial \boldsymbol{\psi}} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \\ &= \mathcal{S}(\boldsymbol{\psi}; \mathbf{y}) \end{aligned}$$

□

Note that in the above proof, the integrals over \mathbf{Z} are in fact generalized integrals (point sums). Note also that for the above proof to hold, the integration over \mathbf{Z} must

commute with the partial derivative $\frac{\partial}{\partial \boldsymbol{\psi}}$. A different proof of result 5.2.1 is given in McLachlan and Krishnan (1996, page 100).

The score functions of the complete-data likelihood are

$$\mathcal{U}_{\boldsymbol{\beta}} = \frac{\partial}{\partial \boldsymbol{\beta}} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) = -\frac{1}{\sigma^2} (\mathbf{X}^T \mathbf{X} \boldsymbol{\beta} - \mathbf{X}^T \mathbf{y}) \quad (5.61)$$

$$\mathcal{U}_{(\sigma^2)} = \frac{\partial}{\partial (\sigma^2)} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^T (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) \quad (5.62)$$

$$\begin{aligned} \mathcal{U}_{\boldsymbol{\phi}} &= \frac{\partial}{\partial \boldsymbol{\phi}} \ln \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) = \sum_{i=1}^s \left(\frac{\partial}{\partial \boldsymbol{\phi}} \mathbf{h}_i^T(\boldsymbol{\phi}) \right) \mathbf{Z}_i^T \mathbf{1}_{n_i} \\ &= \sum_{i=1}^s \left(\frac{\partial}{\partial \boldsymbol{\phi}} \mathbf{h}_i^T(\boldsymbol{\phi}) \right) \mathbf{Z}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i}. \end{aligned} \quad (5.63)$$

Taking the expectation over the distribution \mathbf{Z} given $\ddot{\mathbf{y}}$ and a set of parameters $\boldsymbol{\psi}$, we obtain

$$\frac{\partial}{\partial \boldsymbol{\beta}} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\beta}}] = -\frac{1}{\sigma^2} \left(\widetilde{(\mathbf{X}^T \mathbf{X})} \boldsymbol{\beta} - \widetilde{\mathbf{X}^T} \mathbf{y} \right) \quad (5.64)$$

$$\begin{aligned} \frac{\partial}{\partial (\sigma^2)} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}] \\ &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \left(\mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \widetilde{\mathbf{X}} \boldsymbol{\beta} + \boldsymbol{\beta}^T \widetilde{(\mathbf{X}^T \mathbf{X})} \boldsymbol{\beta} \right) \end{aligned} \quad (5.65)$$

$$\frac{\partial}{\partial \boldsymbol{\phi}} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\phi}}] = \sum_{i=1}^s \left(\frac{\partial}{\partial \boldsymbol{\phi}} \mathbf{h}_i^T(\boldsymbol{\phi}) \right) \widetilde{\mathbf{Z}}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i}. \quad (5.66)$$

Setting these to zero and solving the resulting equations yields maximum likelihood estimators of the parameters for a particular value of $\widetilde{\mathbf{Z}}$, the expected missing data given the observed data. Equations (5.67) to (5.68), below, display the maximum likelihood estimates of the parameters, $\boldsymbol{\beta}$ and σ^2 , based upon a specific value of $\widetilde{\mathbf{Z}}$.

$$\widehat{\boldsymbol{\beta}} = [\widetilde{(\mathbf{X}^T \mathbf{X})}]^{-1} \widetilde{\mathbf{X}}^T \mathbf{y} \quad (5.67)$$

$$\widehat{\sigma^2} = \frac{1}{n} \left(\mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \widetilde{\mathbf{X}} \widehat{\boldsymbol{\beta}} + \widehat{\boldsymbol{\beta}}^T \widetilde{(\mathbf{X}^T \mathbf{X})} \widehat{\boldsymbol{\beta}} \right), \quad (5.68)$$

The MLE for the mixing parameters, $\hat{\phi}$, is more complex. In many applications involving Normal mixtures with equal variance (see McLachlan and Basford 1987, page 38), the MLEs of the mixing parameters take the form given in Equation (5.69).

$$\hat{\mathbf{w}}_i(\phi) = \frac{1}{n_i} \left(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \right) = \frac{1}{n_i} (\tilde{n}_{i1}, \tilde{n}_{i2}, \dots, \tilde{n}_{it}). \quad (5.69)$$

However, in QTL mapping problems, Equation (5.69) does not hold because the w_{ik} are functions of recombination fractions and so they are not functionally independent (example: see Table 5.5). Therefore, any formula for calculating $\hat{\phi}$ will depend on the breeding design and the genetic mapping function.

For backcross design, if we assume Haldane's map function between (but not within) marker intervals, then $\phi = (p_{L1}, p_{L2}, p_{Q1}, p_{Q2}, p_{R1}, p_{R2})^T$. In this situation, the MLEs are found by solving $\frac{\partial}{\partial \phi} \ln \mathcal{L}(\mathbf{y}; \psi) = 0$, and they are as given in equations (5.70) to (5.75) below.

$$\hat{p}_{L1} = \frac{\tilde{n}_{KKLLMM} + \tilde{n}_{KkLlMm}}{n_{KKMM} + n_{KkMm}} \quad (5.70)$$

$$\hat{p}_{L2} = \frac{\tilde{n}_{KKLLMm} + \tilde{n}_{KkLlMM}}{n_{KKMm} + n_{KkMM}} \quad (5.71)$$

$$\hat{p}_{Q1} = \frac{\tilde{n}_{MMQQNN} + \tilde{n}_{MmQqNn}}{n_{MMNN} + n_{MmNn}} \quad (5.72)$$

$$\hat{p}_{Q2} = \frac{\tilde{n}_{MMQQNn} + \tilde{n}_{MmQqNN}}{n_{MMNn} + n_{MmNN}} \quad (5.73)$$

$$\hat{p}_{R1} = \frac{\tilde{n}_{NNRRROO} + \tilde{n}_{NnRrOo}}{n_{NNOO} + n_{NnOo}} \quad (5.74)$$

$$\hat{p}_{R2} = \frac{\tilde{n}_{NNRRROo} + \tilde{n}_{NnRrOO}}{n_{NNOo} + n_{NnOO}} \quad (5.75)$$

Alternatively, if (for the Backcross) we assume Haldane's map function both within and between marker intervals, then $\phi = (p_{L2}, p_{Q2}, p_{R2})^T$. In this case, the MLE's are

found by solving $\frac{\partial}{\partial \phi} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) = 0$ under the constraints given in Equations (5.21) to (5.23). Therefore, in this alternative situation, the MLE's are the solutions of quartic equations in p_{L2} , p_{Q2} and p_{R2} respectively.

The EM maximization procedure for obtaining the MLEs is described below.

Implementation of the EM Algorithm

Step 1: Initialize – select initial values for the elements ($\boldsymbol{\beta}$, σ^2 and ϕ) of the parameter vector $\boldsymbol{\psi}$ (see Section 5.4.1).

Step 2: (E-Step) Using the current estimate of $\boldsymbol{\psi}$, calculate the expected value of the missing data conditioned on the observed data. That is, using current parameter estimates together with Equations (5.48) and (5.49), calculate $\tilde{\mathbf{Z}}_i = E(\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi})$ for $i = 1, \dots, s$ and from these calculate $\tilde{\mathbf{Z}} = E(\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi})$ as in Equation (5.50). Then construct the complete-data model matrix, $\tilde{\mathbf{X}} = [\tilde{\mathbf{Z}}\mathbf{C} \quad \mathbf{X}_2]$.

Step 3: (M-Step) Find new estimates of the parameters by maximizing the conditional expectation of the complete-data log-likelihood given the observed data (see Equations (5.67) to (5.75)).

Step 4: (Update or Terminate) Repeat steps two and three until convergence. \square

5.2.2 The conditional observed information matrix

Under mild regularity conditions, maximum likelihood estimators are asymptotically Normal with mean equal to the true parameter values and variance-covariance matrix equal to the inverse of the Fisher information matrix (Lehmann and Casella, 1998, pages 443-450). The Fisher information matrix is the variance-covariance matrix of the random vector of scores (see Equations (5.61) to (5.63)). Therefore, it

is necessary to determine the conditional observed information matrix for the mixture likelihood because it facilitates calculation of the standard errors of parameter estimates obtained via maximum likelihood estimation.

In the Mixture Modelling Literature, the conditional observed information matrix is often referred to as the ‘observed information matrix’. In this section, we present general formulae for calculating the (conditional) observed information matrix. The formulae presented below have the advantage that, when the number of parameters is large, they are easy to implement in any programming language which allows matrix manipulation.

Like its first derivative, the second derivative of the observed likelihood with respect to the parameter $\boldsymbol{\psi}$, may also be written as a function of the score of the complete-data likelihood. This leads to a formula for the observed information, $\mathcal{I}(\boldsymbol{\psi}; \mathbf{y})$, in terms of the complete-data information and the missing information.

$$\begin{aligned}
 \mathcal{I}(\boldsymbol{\psi}; \mathbf{y}) &= -\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \\
 &= -\frac{\partial}{\partial \boldsymbol{\psi}} \left\{ \frac{\partial}{\partial \boldsymbol{\psi}^T} \ln \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \right\} \\
 &= -\frac{\partial}{\partial \boldsymbol{\psi}} \left\{ \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \left(\frac{\partial}{\partial \boldsymbol{\psi}^T} \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \right) \right\} \\
 &= \mathcal{S}(\boldsymbol{\psi}; \mathbf{y}) \mathcal{S}^T(\boldsymbol{\psi}; \mathbf{y}) - \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \left(\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \right) \quad (5.76)
 \end{aligned}$$

Therefore, using Result 5.2.1 once again,

$$\mathcal{I}(\boldsymbol{\psi}; \mathbf{y}) = (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}}])(E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}}])^T - \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \left(\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \right). \quad (5.77)$$

Likewise

$$\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z}) = \mathcal{S}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z}) \mathcal{S}_c^T(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z}) - \frac{1}{\mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi})} \left(\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) \right). \quad (5.78)$$

$$\begin{aligned}
& \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \left(\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) \right) \\
&= \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \left(\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \int \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) d\mathbf{Z} \right) \\
&= \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \int \frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) d\mathbf{Z} \\
&= \frac{1}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \int \frac{\mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi})}{\mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi})} \left(\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) \right) d\mathbf{Z} \\
&= \int \frac{\mathcal{L}_m(\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi})}{\mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi})} \left(\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) \right) d\mathbf{Z}, \\
&\quad \text{where } \mathcal{L}_m(\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}) = \frac{\mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi})}{\mathcal{L}(\mathbf{y}; \boldsymbol{\psi})} \\
&= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} \left[\frac{1}{\mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi})} \left(\frac{\partial^2}{\partial \boldsymbol{\psi} \partial \boldsymbol{\psi}^T} \mathcal{L}_c(\mathbf{y}, \mathbf{Z}; \boldsymbol{\psi}) \right) \right] \\
&= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{S}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z}) \mathcal{S}_c^T(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z}) - \mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z})] \text{ from Equation (5.78),} \\
&= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{S}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z}) \mathcal{S}_c^T(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z})] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z})] \\
&= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{U}_\psi \mathcal{U}_\psi^T] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [-\mathcal{U}_\psi \boldsymbol{\psi}] \tag{5.79}
\end{aligned}$$

Define

$$\text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{U}_\psi, \mathcal{U}_\psi] = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{U}_\psi \mathcal{U}_\psi^T] - (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{U}_\psi]) (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{U}_\psi])^T. \tag{5.80}$$

To use similar notation to that of McLachlan and Krishnan (1996), let

$$\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}) = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z})] = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [-\mathcal{U}_\psi \boldsymbol{\psi}], \tag{5.81}$$

$$\mathcal{I}_m(\boldsymbol{\psi}; \mathbf{y}) = \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{U}_\psi, \mathcal{U}_\psi]. \tag{5.82}$$

Substituting the results of Equations (5.79) to (5.82) into the above expression for the observed information, $\mathcal{I}(\boldsymbol{\psi}; \mathbf{y})$, yields

$$\begin{aligned}
\mathcal{I}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [-\mathcal{U}_\psi \boldsymbol{\psi}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}} [\mathcal{U}_\psi, \mathcal{U}_\psi] \\
&= \mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}) - \mathcal{I}_m(\boldsymbol{\psi}; \mathbf{y}). \tag{5.83}
\end{aligned}$$

The term $\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y})$ represents the amount of information about $\boldsymbol{\psi}$ that, given the observed data, is expected from observation of the complete data if the latter were available. On the other hand, $\mathcal{I}_m(\boldsymbol{\psi}; \mathbf{y})$ is the expected amount of information about $\boldsymbol{\psi}$ that, given the observed data, is associated with the conditional likelihood of the missing data (see Louis (1982), McLachlan and Krishnan (1996, Chap. 3)).

Element-by-element evaluation of the observed information matrix can be prohibitively tedious when a mixture model depends on a large number of parameters. For example, if we fit no extra cofactors, then our backcross model has nine parameters and so the information matrix is a 9×9 matrix comprising 81 elements. We note that the information matrix is symmetric and therefore we have $9(9+1)/2 = 45$ elements to calculate separately. This is still a fairly large number of elements, and so incorporating such calculations into any computer program would prove to be both time-consuming and susceptible to typographical errors. Element-by-element evaluation of the information matrix quickly becomes implausible when several extra cofactors are included.

Below, we provide formulae to calculate the observed information matrix, without separately evaluating each of its elements. We show that, irrespective of the number of parameters in a mixture model, a maximum of ten matrix expressions (blocks) need to be calculated when evaluating the observed information matrix. Only six blocks are needed if no extra cofactors are fitted. For the Fisher Information matrix, $\mathcal{I}(\boldsymbol{\psi}) = E_{\mathbf{y}; \boldsymbol{\psi}} [\mathcal{I}(\boldsymbol{\psi}; \mathbf{y})]$, either three or five blocks need to be calculated depending on whether extra cofactors are included in the model. The first and second partial derivatives of the mixing proportions are simple to calculate because they do not depend on the data. The formulae presented here avoid element-by-element calculations by directly operating on matrices which are already available from the model fitting step, and by directly operating on matrices containing only the first and second partial derivatives of the mixing proportions.

In the discussion below, we use the following notation:

- (1) For $\mathbf{C}\mathbf{b}$, the column vector of means, we write: $\boldsymbol{\mu} = \mathbf{C}\mathbf{b}$.
- (2) For a row vector \mathbf{v}^T , we use $\text{diag}(\mathbf{v}^T)$ to denote the diagonal matrix whose i^{th} diagonal element is given by the i^{th} element of \mathbf{v}^T .
- (3) The number of (marker) groupings on which we condition is denoted by s and we note that $s = 16$ for a backcross design, while $s = 81$ for a F2 design.

First, we express the observed information matrix as the partitioned matrix given in Equation (5.84) below.

$$\mathcal{I}(\boldsymbol{\psi}; \mathbf{y}) = \begin{bmatrix} \mathcal{I}_{\mathbf{bb}}(\boldsymbol{\psi}; \mathbf{y}) & \mathcal{I}_{\mathbf{bb}^*}(\boldsymbol{\psi}; \mathbf{y}) & \mathcal{I}_{\mathbf{b}(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) & \mathcal{I}_{\mathbf{b}\phi}(\boldsymbol{\psi}; \mathbf{y}) \\ [\mathcal{I}_{\mathbf{bb}^*}(\boldsymbol{\psi}; \mathbf{y})]^T & \mathcal{I}_{\mathbf{b}^*\mathbf{b}^*}(\boldsymbol{\psi}; \mathbf{y}) & \mathcal{I}_{\mathbf{b}^*(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) & \mathcal{I}_{\mathbf{b}^*\phi}(\boldsymbol{\psi}; \mathbf{y}) \\ [\mathcal{I}_{\mathbf{b}(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y})]^T & [\mathcal{I}_{\mathbf{b}^*(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y})]^T & \mathcal{I}_{(\sigma^2)(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) & \mathcal{I}_{(\sigma^2)\phi}(\boldsymbol{\psi}; \mathbf{y}) \\ [\mathcal{I}_{\mathbf{b}\phi}(\boldsymbol{\psi}; \mathbf{y})]^T & [\mathcal{I}_{\mathbf{b}^*\phi}(\boldsymbol{\psi}; \mathbf{y})]^T & [\mathcal{I}_{(\sigma^2)\phi}(\boldsymbol{\psi}; \mathbf{y})]^T & \mathcal{I}_{\phi\phi}(\boldsymbol{\psi}; \mathbf{y}) \end{bmatrix} \quad (5.84)$$

Then we calculate each partition using appropriate functions of certain matrices that are available from the E-M procedure.

Equations (5.85) to (5.94) display exact formulae for calculating the ten distinct blocks that comprise the upper triangle of the (symmetric) conditional information matrix.

The matrix expressions representing each component were found by directly applying the definition of the observed information, together with the rules of expectation and the rules of matrix addition and matrix equality. The proofs are provided in Chapter 6, which can be regarded as a technical appendix to the current chapter.

$$\begin{aligned}
\mathcal{I}_{\mathbf{bb}}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{bb}}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}, \mathcal{U}_{\mathbf{b}}] \\
&= \frac{1}{\sigma^2} \mathbf{C}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \mathbf{C} \\
&\quad - \frac{1}{\sigma^4} \mathbf{C}^T \left[\text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) \right. \\
&\quad \left. + \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T) - 2 \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) \right) \right. \\
&\quad \left. - \left(\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right) \left(\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right)^T \right] \mathbf{C} \quad (5.85)
\end{aligned}$$

$$\begin{aligned}
\mathcal{I}_{(\sigma^2)(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{(\sigma^2)(\sigma^2)}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}, \mathcal{U}_{(\sigma^2)}] \\
&= \frac{1}{\sigma^6} \left(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}} - 2 \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} \boldsymbol{\mu} + \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \right) - \frac{n}{2\sigma^4} \\
&\quad - \frac{1}{4\sigma^8} \boldsymbol{\mu}^T \left[4 \text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) - 4 \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \right. \\
&\quad \left. + 4 \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \right) \right. \\
&\quad \left. + \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \right) \text{diag}(\boldsymbol{\mu}^T) \right] \boldsymbol{\mu} \quad (5.86)
\end{aligned}$$

$$\begin{aligned}
\mathcal{I}_{\phi\phi}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\phi\phi}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\phi}, \mathcal{U}_{\phi}] \\
&= - \sum_{i=1}^s \left(\frac{\partial^2}{\partial \phi \partial \phi^T} \mathbf{h}_i^T(\phi) \right) \tilde{\mathbf{Z}}_i^T \mathbf{1}_{n_i} \\
&\quad - \sum_{i=1}^s \sum_{i'=1}^s \left[\left(\frac{\partial}{\partial \phi} \mathbf{h}_i^T(\phi) \right) \left(\text{diag}(\mathbf{1}_{n_{i'}}^T \boldsymbol{\Delta}_{i'} \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i) \right. \right. \\
&\quad \left. \left. - \tilde{\mathbf{Z}}_i^T \boldsymbol{\Delta}_i \boldsymbol{\Delta}_{i'}^T \tilde{\mathbf{Z}}_{i'} \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_{i'}^T(\phi) \right) \right]. \quad (5.87)
\end{aligned}$$

$$\begin{aligned}
\mathcal{I}_{\mathbf{b}(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}(\sigma^2)}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}, \mathcal{U}_{(\sigma^2)}] \\
&= -\frac{1}{\sigma^4} \mathbf{C}^T \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} - \tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \right) \\
&\quad - \frac{1}{2\sigma^6} \mathbf{C}^T \left[2 \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \right) \right. \\
&\quad \quad - 2 \text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) + 2 \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \\
&\quad \quad - \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \right) \text{diag}(\boldsymbol{\mu}^T) \\
&\quad \quad \left. + \left(\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \right) \text{diag}(\boldsymbol{\mu}^T) \right] \boldsymbol{\mu} \quad (5.88)
\end{aligned}$$

$$\begin{aligned}
\mathcal{I}_{\mathbf{b}\phi}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}\phi}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}, \mathcal{U}_{\phi}] \\
&= \frac{1}{\sigma^2} \mathbf{C}^T \sum_{i=1}^s \left[\left(\text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i \right. \right. \\
&\quad \quad \left. \left. - \text{diag}(\ddot{\mathbf{y}}^T \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i) + \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right] \quad (5.89)
\end{aligned}$$

$$\begin{aligned}
\mathcal{I}_{(\sigma^2)\phi}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{(\sigma^2)\phi}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}, \mathcal{U}_{\phi}] \\
&= -\frac{1}{2\sigma^4} \boldsymbol{\mu}^T \sum_{i=1}^s \left[\left(\text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i \right. \right. \\
&\quad \quad \left. \left. - 2 \text{diag}(\ddot{\mathbf{y}}^T \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i) + 2 \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right] \quad (5.90)
\end{aligned}$$

The components of the observed information matrix that are associated with the extra cofactors are given below.

$$\begin{aligned}
\mathcal{I}_{\mathbf{b}\mathbf{b}^*}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}\mathbf{b}^*}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}, \mathcal{U}_{\mathbf{b}^*}] \\
&= \frac{1}{\sigma^2} \mathbf{C}^T \tilde{\mathbf{Z}}^T \mathbf{X}_2 - \frac{1}{\sigma^4} \mathbf{C}^T \left[\text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right) \right. \\
&\quad \quad \left. - \left(\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \right] \mathbf{X}_2 \quad (5.91)
\end{aligned}$$

$$\begin{aligned}
\mathcal{I}_{\mathbf{b}^*\mathbf{b}^*}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*\mathbf{b}^*}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}, \mathcal{U}_{\mathbf{b}^*}] \\
&= \frac{1}{\sigma^2} \mathbf{X}_2^T \mathbf{X}_2 - \frac{1}{\sigma^4} \mathbf{X}_2^T \left[\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T) \right. \\
&\quad \left. - \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \right] \mathbf{X}_2
\end{aligned} \tag{5.92}$$

$$\begin{aligned}
\mathcal{I}_{\mathbf{b}^*(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*(\sigma^2)}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}, \mathcal{U}_{(\sigma^2)}] \\
&= \frac{1}{\sigma^4} \mathbf{X}_2^T (\tilde{\mathbf{Z}}\boldsymbol{\mu} - \ddot{\mathbf{y}}) - \frac{1}{2\sigma^6} \mathbf{X}_2^T \left[2 \text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}} \right. \\
&\quad - 2 \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}} \\
&\quad \left. - \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \boldsymbol{\mu} + \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \boldsymbol{\mu} \right]
\end{aligned} \tag{5.93}$$

$$\begin{aligned}
\mathcal{I}_{\mathbf{b}^*\phi}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*\phi}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}, \mathcal{U}_{\phi}] \\
&= \frac{1}{\sigma^2} \mathbf{X}_2^T \left[\sum_{i=1}^s \left[\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \tilde{\mathbf{Z}} \right. \right. \\
&\quad \left. \left. - \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \right] \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right].
\end{aligned} \tag{5.94}$$

These calculations are quite general and they may be used to obtain the observed information matrix for a wide variety linear models involving mixtures of Normals. Note however that, these formulae for the conditional observed information matrix were only derived as a first step to obtaining formulae for the Fisher information matrix. The conditional observed information will not generally be a good approximation of the Fisher information unless the sample size is large (Basford *et al.* (1997)). Moreover, it is possible for the conditional information matrix to be negative definite, whereas the Fisher information matrix is always non-negative definite. Therefore, in practice, the Fisher information matrix formulae given in the next section should be used when calculating the covariance matrix of the model parameters.

5.2.3 The Fisher information matrix

In this section, we calculate the Fisher information for our Normal mixture model.

Let $\bar{\mathbf{Z}} = E_{\mathbf{y}; \psi}[\tilde{\mathbf{Z}}]$. Then $\bar{\mathbf{Z}}$ is the expected value of the missing data over all possible observations and

$$\bar{\mathbf{Z}} = \begin{bmatrix} \mathbf{1}_{n_1} \mathbf{w}_1^T(\phi) \\ \mathbf{1}_{n_2} \mathbf{w}_2^T(\phi) \\ \vdots \\ \mathbf{1}_{n_s} \mathbf{w}_s^T(\phi) \end{bmatrix}. \quad (5.95)$$

By definition, the Fisher information is

$$\begin{aligned} \mathcal{I}(\psi) &= E_{\mathbf{y}; \psi}[\mathcal{I}(\psi; \mathbf{y})] \\ &= E_{\mathbf{y}; \psi}[\mathcal{S}(\psi; \mathbf{y})\mathcal{S}^T(\psi; \mathbf{y})]. \end{aligned} \quad (5.96)$$

The components of $\mathcal{I}(\psi; \mathbf{y})$ are given, explicitly, in equations (5.85) to (5.94). The Fisher information matrix is calculated by taking the expectation over \mathbf{y} of the expressions in equations (5.85) to (5.94). These expectations were calculated using Equations (6.73) to (6.94) which are provided in Section 6.7. The resulting formula for the Fisher information matrix of our mixture likelihood is given in (5.97) below.

$$\mathcal{I}(\psi) = \begin{bmatrix} \frac{1}{\sigma^2} \mathbf{C}^T \text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}}) \mathbf{C} & \frac{1}{\sigma^2} \mathbf{C}^T \bar{\mathbf{Z}}^T \mathbf{X}_2 & \mathbf{0} & \mathbf{0} \\ \frac{1}{\sigma^2} \mathbf{X}_2^T \bar{\mathbf{Z}} \mathbf{C} & \frac{1}{\sigma^2} \mathbf{X}_2^T \mathbf{X}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{n}{2\sigma^4} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\sum_{i=1}^s \left(\frac{\partial^2}{\partial \phi \partial \phi^T} \mathbf{h}_i^T(\phi) \right) \bar{\mathbf{Z}}_i^T \mathbf{1}_{n_i} \end{bmatrix} \quad (5.97)$$

It is interesting to note that the formulae for the components of the expected information matrix turn out to be vastly simpler than those for the observed information. As an illustration of how one might implement these formulae in practice, R program code to calculate both the observed and the expected information matrices (for seven different models) is given in Appendix B.4.

5.3 Hypothesis testing

This section begins by discussing the options that are available for testing hypotheses about parameters in our mixture model. Then it outlines our chosen hypothesis testing strategy.

From an interval mapping perspective, we are mainly interested in parameters of \mathbf{b} which are associated with the QTL in the testing interval. For the backcross, there are only two genotypes at each locus. Therefore, only one contrast can be fitted to capture the main effects at each locus. If one imagines that the main effects have additive and dominance components which are both non-zero, then clearly these cannot be separated in the backcross, because we cannot estimate two unknowns from one equation. As previously shown in Equation (5.6), the expected value of b_Q is equal to $(a_{QQ} + 2d_{QQ})$ for the backcross. For the F2, there are three genotypes at each locus, so one may fit at most two main effects, and it is possible to separate additive and dominance components (see also Sections 4.1.3 and 4.1.4).

Is there a QTL interior to the testing interval? To answer this question for the backcross model we need to test the pair of hypotheses

$$H_1 : (b_Q \neq 0) \text{ and } (p_{Q2} \neq 0) \text{ and } (p_{Q2} \neq 1),$$

$$H_0 : (b_Q = 0) \text{ or } (p_{Q2} = 0) \text{ or } (p_{Q2} = 1).$$

Note that if $(p_{Q2} = 0)$ or $(p_{Q2} = 1)$, then the QTL is at one of the two flanking markers. Moreover

$$(p_{Q2} = 0) \Leftrightarrow (r_{MQ} = r_{MN}) \text{ and } (p_{Q2} = 1) \Leftrightarrow (r_{MQ} = 0).$$

We have a composite hypothesis test for QTL effect and location. In applications such as Marker-Assisted Selection (MAS), where exact location might not be so important, the simple hypothesis test for QTL effect:

$$H_{0b} : b_Q = 0 \text{ versus } H_{1b} : b_Q \neq 0$$

might also be of interest. Such a test would indicate if there is a QTL tightly linked to the markers. However, if we want to make statements about whether Q is interior to the testing interval, then the test for effect should be used in conjunction with a test for position.

Consider the behaviour of the EM algorithm when Q is included in the model and the null hypothesis (that Q has zero effect) is in fact true. In such situations the EM algorithm tends to move from initial values towards parameter space boundaries either by causing some of the estimated means, μ_k , to become equal or by moving some of the mixing proportions, w_{ik} , towards zero (Lesperance and Lindsay (2001)).

This behaviour means that when there is no QTL interior to the interval $M - N$, the EM could generate a value of b_Q that is close to zero. Alternatively, it could generate a value of b_Q that is significantly greater than zero while simultaneously pushing the location of the QTL (Q) towards either end of the interval. For this reason, the composite hypothesis test for both effect and location will be more robust to false detections than the test for effect alone. Simulations indicated (see Chapter 7) that this consideration is more important for reducing ghosting in Composite Interval Mapping (CIM) than in Robust Interval Mapping Version 1 (RIM1). The RIM1 model displays low ghosting irrespective of whether we test for effect only (H_{0b}) or for both effects and location (H_0), whereas CIM only displays low ghosting when H_0 is used for the null hypothesis.

Estimators of other parameters such as b_L and b_R are also interesting when scanning a linkage group to search for QTL because they allow us to perform informal checks, for robustness and consistency, by comparing the values of estimators as the testing interval is moved from one interval to the next.

In likelihood-based models, classical inference about model parameters takes the sampling distributions of suitable statistics under the null hypothesis, together with

chosen significance levels, to construct threshold values for acceptance of the null hypothesis. The extent to which observed data supports acceptance or rejection of the null hypothesis is then determined by estimating the chosen statistic and comparing its value to the threshold values or to their associated rejection regions. The preferred tests are likelihood ratio tests, Lagrangian multiplier or score tests and Wald t-tests (Cox and Hinkley, 1974). These tests are asymptotically equivalent and their equivalence is based on a quadratic Taylor-series expansion of the score function.

The models under consideration (CIM and RIM1) are mixtures of univariate normals having equal variances. Therefore, applying the results of Redner and Walker (1984), we see that under alternative hypothesis (H_1) the MLE $\hat{\psi}$ obtained from the EM algorithm is consistent for ψ in the sense that as the sample size approaches infinity, the MLE converges with probability one to the true parameter value (see also Basford and McLachlan, 1985; McLachlan and Krishnan, 1996). Moreover, under H_1 , the maximum likelihood estimators for the parameters of these mixture models have an asymptotic Normal distribution with mean equal to the true parameter vector, and covariance matrix equal to the inverse of the Fisher information matrix. This result comes from established asymptotic theory (Redner, 1981; Redner and Walker, 1984; Titterington *et al.*, 1985, pages 91-93; McLachlan and Krishnan, 1996, pages 111-113).

The null distribution of the mixture likelihood is complicated because the null model does not conform to the regularity conditions which are required for the score statistic to be asymptotically normal (Titterington, 1981; Ghosh and Sen, 1985). This departure from regularity has two main causes.

1. Under the null hypothesis, the mixing parameters may lie on the boundary of the parameter space.

2. The parameters are not identifiable even when the class of mixtures is identifiable. For example, the three statements $p_{Q2} = 0$, $p_{Q2} = 1$, and $b_Q = 0$ are equivalent because each statement implies that the marker means do not depend on the genotypes at locus Q . Consequently, the same probability density function may be generated by different parameter values.

The breakdown in regularity means that the likelihood ratio test statistic,

$$\Lambda = -2(\ln \mathcal{L}(\mathbf{y}; \hat{\boldsymbol{\psi}}_0) - \ln \mathcal{L}(\mathbf{y}; \hat{\boldsymbol{\psi}})),$$

does not have the standard asymptotic chi-square null distribution with degrees of freedom equal to the difference in the number of parameters in the two hypotheses (see McLachlan and Basford, 1987, pages 21-29).

Various researchers have shown that the true distribution of the LRT involves a Gaussian stochastic process. This distribution is difficult to calculate in practice. Chen *et al.* (2001) reviewed the work of these researchers, summarised the properties of this Gaussian process and outlined the difficulties of calculating its distribution. As an alternative strategy for testing for homogeneity in finite mixture models, they also proposed a modified likelihood ratio test which has simpler asymptotic properties.

Despite the aforementioned concerns, the standard chi-square distribution is often used with the LRT in QTL mapping applications based on mixture distributions. In some applications researchers were able to successfully detect QTL despite using this less than ideal approximation (examples: Jansen and Stam, 1994; Zeng, 1994). The main disadvantage of using the the standard chi-square distribution with the mixture LRT is that the rate of false detections can be unacceptably high.

An alternative option is to use empirical estimators obtained by data re-sampling in order to approximate the distributions of the LRT or of the score statistic. Churchill and Doerge (1994) applied permutation tests or re-sampling without replacement to

map QTL. Visscher *et al.* (1996b) used bootstrapping or re-sampling with replacement to map QTL. In the bootstrap approach to hypothesis testing, the empirical cumulative distribution function of the bootstrap estimators of a test statistic is used to approximate its true distribution. Care must be taken when using the bootstrap approach because Bickel and Freedman (1981) and Swanepoel (1986) have shown that erroneous results are possible when the bootstrap distribution is not a consistent estimator of the true distribution. The question of how many bootstrap samples to take is also important. Beran and Ducharme (1991) suggested that between 1,000 and 10,000 bootstraps would be adequate.

Another approach is to use the asymptotic theory of Self and Liang (1987) when parameters are on the boundary of the parameter space. Self and Liang (1987) prove, in the presence of identifiability, that if none of the nuisance parameters are on the boundary and some of the main parameters are on the boundary under the null hypothesis, then the LRT is distributed as a mixture of chi-squares. Under these conditions, Self and Liang (1987) found that the large-sample distribution of the LRT statistic is approximately equal to

$$\sum_{i=1}^{2^q} \chi_{r-\nu_i}^2 P\left(\mathbf{L}_i \mathbf{V}^{-\frac{1}{2}} \mathbf{P}^T \mathcal{S}(\boldsymbol{\psi}) > 0\right)$$

where r , q , ν_i , \mathbf{L}_i , \mathbf{V} , \mathbf{P} , and $\mathcal{S}(\boldsymbol{\psi})$ are as described below.

1. Suppose that the parameter vector $\boldsymbol{\psi}$ contains p elements. Suppose also that, under the null hypothesis the first r components of $\boldsymbol{\psi}$ are explicitly specified. Then, under the null hypothesis, $\boldsymbol{\psi}$ is restricted to $\boldsymbol{\psi}_0$ with its first r values specified ($r \leq p$).
2. For $q \leq r$, suppose that the first q components of $\boldsymbol{\psi}_0$ lie on the parameter space boundary.
3. Under the alternative hypothesis, the MLE's of the first q parameters of $\boldsymbol{\psi}$

may or may not lie on the boundary. Therefore, there are 2^q configurations for $\boldsymbol{\psi}$, where a configuration indicates which of the first q parameters lie on the boundary, and i is used to index these configurations.

4. The term ν_i represents the number of elements (among the first q elements of $\boldsymbol{\psi}$) that are on the boundary in the i^{th} configuration. When $\nu_i < r$ the expression $\chi_{r-\nu_i}^2$ denotes the chi-square distribution having $r - \nu_i$ degrees of freedom. We take χ_0^2 to be the distribution with a point mass of one at zero.
5. In the i^{th} configuration, let \mathbf{B}_i denote the $p \times p$ diagonal matrix, with its j^{th} diagonal element equal to one if the j^{th} coordinate of $\boldsymbol{\psi}$ is on the boundary, and equal to zero otherwise. Then construct the orthogonal projection matrix \mathbf{P}_i where

$$\mathbf{P}_i = \mathbf{I}_p - \mathbf{B}_i \mathcal{I}^{-1}(\boldsymbol{\psi}_0) [\mathbf{B}_i^T \mathcal{I}^{-1}(\boldsymbol{\psi}_0) \mathbf{B}_i]^+ \mathbf{B}_i^T.$$

Let $\mathbf{0}$ be a $q \times (p - q)$ matrix with all elements equal to zero. Then, \mathbf{L}_i is the $q \times p$ matrix given by

$$\mathbf{L}_i = [\mathbf{I}_q \quad \mathbf{0}](\mathbf{P}_i - \mathbf{B}_i).$$

6. The matrices \mathbf{P} and \mathbf{V} are such that \mathbf{PVP}^T is the spectral decomposition of $\mathcal{I}(\boldsymbol{\psi}_0)$, where $\mathcal{I}(\boldsymbol{\psi}_0)$ is the Fisher information matrix evaluated under the null hypothesis.
7. The object $\mathcal{S}(\boldsymbol{\psi})$ is the vector of scores and $\mathbf{V}^{-\frac{1}{2}} \mathbf{P}^T \mathcal{S}(\boldsymbol{\psi})$ is its Mahalanobis transformation when $\boldsymbol{\psi} = \boldsymbol{\psi}_0$. The limiting distribution of this Mahalanobis transformation is that of a Gaussian random vector having mean zero and identity covariance matrix. This allows us to calculate the mixing probabilities for this mixture of chi-squares.

If one or more nuisance parameters lie on the boundary, then the asymptotic distribution of the LRT will not necessarily be a mixture of chi-squared random variables

unless parameters of interest and these nuisance parameters are uncorrelated (Self and Liang, 1987; Schoenberg, 2001). In the case of our RIM1 model, it is possible for nuisance parameters such as p_{R2} and p_{L2} to fall on the boundary of the parameter space. However, the information matrix is block diagonal with respect to β , σ and ϕ . Additionally, if Haldane's addition formula is assumed everywhere, then p_{R2} , p_{Q2} , and p_{L2} (the components of ϕ) will be uncorrelated with each other.

The results of Self and Liang (1987) were derived in the context of samples of independent identically distributed random variables. Vu and Zhou (1997) and Andrews (1999) extended these results to allow for more relaxed assumptions.

Note also that the results of Self and Liang (1987) require identifiability. Normal mixture distributions do not satisfy this requirement. In fact, the common feature of the results of Self and Liang (1987), Vu and Zhou (1997) and Andrews (1999) is that they all require the existence and consistency of the MLE for ψ_0 . Lack of identifiability is often a barrier to obtaining a consistent estimator for ψ_0 (see Lehmann and Casella, 1998, page 443).

The fact that our model is unidentified does not put a caveat on proceeding. Breusch (1986), for example, pointed out that not only is inference possible with unidentified models but that such inference is often undertaken in the analysis of many common systems.

Breusch (1986) provided a synopsis of hypothesis testing strategies that are suitable for use with different types of unidentified models. His synopsis included the works of Silvey (1959), Aitchison and Silvey (1960) and Davies (1977).

Davies (1977, 1987) recommended the use of the maximum of score statistics when a nuisance parameter is only present under the alternative hypothesis. Chang *et al.* (2003) also developed score tests for QTL. Chang *et al.* (2003) appealed to Gaussian stochastic processes that were previously described in Chen and Chen (1998a,b) for testing homogeneity in mixture models. They showed that, for large samples, the

maximum of the square of the score statistic has a null distribution which is approximately equal to the distribution of the maximum of the square of a well-defined Gaussian process. Then they used simulations to compute this distribution. For backcross samples, Chang *et al.* (2003) found that their method yielded threshold values that were similar to those obtained by the permutation method of Churchill and Doerge (1994). However, the score statistic approach of Chang *et al.* (2003) had a considerable advantage in reducing computing times.

When choosing a method for testing the hypothesis H_0 versus H_1 , it is important to consider the fact that, for our mixture distribution, the null hypothesis is not simply nested within the alternative. Equation (5.97) shows that the Fisher information matrix is block diagonal. This means that we can carry out tests for β and ϕ independently. Therefore, we propose a type of sequential test, which begins with a hypothesis test for whether the QTL effect \hat{b}_Q is significantly different from zero. That is, we first test $H_{0b} : b_Q = 0$. If a significant QTL effect is found (H_{0b} is rejected), then hypothesis testing continues with another test to determine whether we have significant evidence that the QTL is strictly interior to the testing interval. If H_{0b} is rejected, we construct an approximate interval test based on the two null hypotheses:

$$H_{0m} : p_{Q2} = 1, \text{ and } H_{0n} : p_{Q2} = 0.$$

Note that if H_{0b} is accepted, then there is no need to carry out tests on p_{Q2} and we simply declare that there is not enough evidence for a QTL in the interval $M-N$.

We obtain the variances of the parameter estimates from the inverse of the Fisher information matrix given in Equation (5.97), and we use the test statistics given in Equations (5.98) to (5.102).

$$T_1 = T_1(\hat{b}_Q) = \frac{\hat{b}_Q - 0}{\sqrt{\text{var}(\hat{b}_Q)}} \quad (5.98)$$

Likewise, define J_m and J_n as tests statistics for whether Q is located at markers M and N respectively.

$$J_m = J_m(\hat{p}_{Q2}) = \frac{\hat{p}_{Q2} - 1}{\sqrt{\text{var}(\hat{p}_{Q2})}} \quad (5.99)$$

$$J_n = J_n(\hat{p}_{Q2}) = \frac{\hat{p}_{Q2} - 0}{\sqrt{\text{var}(\hat{p}_{Q2})}} \quad (5.100)$$

Also define the following vector-valued test statistics

$$J_c = J_c(\hat{p}_{Q2}) = (J_m, J_n) \quad (5.101)$$

$$J_1 = J_1(\hat{b}_Q, \hat{p}_{Q2}) = (T_1, J_m, J_n) = (T_1, J_c) \quad (5.102)$$

To implement the hypothesis tests, we need some strategy for calculating or approximating the distributions of T_1 , J_m and J_n under the respective null hypotheses H_{0b} , H_{0m} and H_{0n} . The asymptotic distribution of T_1 is the least problematic of the three required distributions.

Under H_{0b} , the parameter b_Q is the only restricted parameter and it is strictly interior to the parameter space boundaries. All mixing proportions are constrained to be greater than zero in the mixture model. Therefore, the EM maximization process never allows the probabilities p_{L2} , p_{Q2} and p_{R2} to fall upon parameter space boundaries, although they can become arbitrarily close to it. The fact that the information matrix is block diagonal with separate blocks corresponding to β and ϕ means that the proximity of elements of $\hat{\phi}$ to the parameter space boundary does not affect the asymptotic distribution of $\hat{\beta}$ (Andrews, 1999; Schoenberg, 2001). Therefore, for large samples, the statistic T_1 will be almost Standard Normal if the true value of b_Q is equal to zero.

$$T_1 \sim N(0, 1) \text{ when } H_{0b} \text{ is true.}$$

The p-value for a test of $b_Q = 0$ versus $b_Q \neq 0$ is given by

$$\text{p-value of } T_1 = 2 P(\bullet > |T_1|). \quad (5.103)$$

If the p-value of T_1 is less than the chosen significance level, then H_{0b} is rejected, and a QTL Q associated with the interval $M - N$ is detected.

It is more problematic to ascertain the asymptotic distributions of J_m under H_{0m} , and J_n under H_{0n} . There are two main problems:

- If $b_Q = 0$, then p_{Q2} can never be consistently estimated. However, it is reasonable to expect that this problem will be mitigated by the fact that we will only use J_m and J_n after finding that b_Q is significantly different from zero.
- When H_{0m} or H_{0n} are true, the parameter p_{Q2} lies on the boundary of the parameter space.

Even when $b_Q > 0$, consistency of the MLEs under H_{0m} and H_{0n} is not guaranteed. Nevertheless, to construct *rough tests*, we appeal to Self and Liang (1987), Andrews (1999) and Schoenberg (2001), and to the fact that p_{Q2} is uncorrelated with the other elements of ψ . We take the asymptotic distribution of J_m under H_{0m} to be a 50:50 mixture of a degenerate distribution (point mass 1 at zero) and a left-truncated Standard Normal distribution (truncated to the left of zero), and assume that

$$\text{when } H_{0m} \text{ is true, } P(\bullet \leq J_m) \approx \begin{cases} 1/2 & \text{if } J_m = 0 \\ 1/2 + \Phi(x) \Big|_{x=0}^{x=J_m} & \text{if } J_m > 0, \end{cases} \quad (5.104)$$

where $\Phi(x)$ is the Standard Normal distribution function.

Similarly, we use a right-truncated Standard Normal distribution and assume that

$$\text{when } H_{0n} \text{ is true, } P(\bullet \leq J_n) \approx \begin{cases} \Phi(x) \Big|_{x=-\infty}^{x=J_n} & \text{if } J_n < 0. \\ 1 & \text{if } J_n = 0 \end{cases} \quad (5.105)$$

The p-value for a test of whether Q is interior to the interval $M - N$ is calculated as follows.

$$\begin{aligned} \text{p-value of } J_c &= \text{p-value of } J_m + \text{p-value of } J_n \\ &= P(\bullet < J_m) + P(\bullet > J_n) \end{aligned} \quad (5.106)$$

If T_1 is significantly different from zero, and the p-value of J_c is less than half of the chosen significance level, then there is evidence for a linked QTL which is strictly interior to the testing interval. We use half of the chosen significance level in order to implement a Bonferroni correction for multiple testing on p_{Q2} with J_c . The p-value of J_1 is taken to be the maximum of the p-values of T_1 and J_c .

Despite the fact that these are rough tests, the results in Chapter 7 demonstrate that these tests have good power to detect QTL and that they are dramatically more resistant to ghosting than the Chi-square LRT.

For comparison purposes, new permutation tests are also proposed. Consider the statistics T_2 and J_2 where

$$T_2 = b_Q^2 \quad (5.107)$$

$$J_2 = p_{Q2}(1 - p_{Q2})(b_Q^2). \quad (5.108)$$

The right tail of the empirical null-distribution of T_2 can be used to calculate p-values for a test of whether $b_Q = 0$, thereby giving an alternative to the asymptotic T_1 test described above. Likewise, a test based on the empirical null-distribution of J_2 can be used rather than the ‘two-step’ J_1 test described above.

The permutation method of Churchill and Doerge (1994) involves randomly shuffling the trait values among individuals, while retaining each individual’s genetic data. This method is only appropriate when there is a single explanatory variable – for example, fitting a single QTL with no cofactors.

Permuting the sample as per Churchill-Doerge will destroy any association between the trait and all genotypes. Still, the Churchill-Doerge method might not give the correct null situation because it will destroy *all* associations, leading to a situation where H_0 is equivalent to ‘no QTL anywhere’, when the true H_0 really should be ‘no QTL interior to $M - N$ ’ (which means that QTL could be outside the interval).

Rather than shuffling the trait values, it is better to randomise the covariate of

interest among the subjects. This randomisation method is discussed in Manly (1997, Chapter 8) in the context of constructing a randomisation test for a coefficient in a multiple regression. For our QTL mapping problem, we implement this method by permuting the two-locus ‘ MN ’ marker genotypes within each group defined by the two-locus ‘ KO ’ genotypes.

For example, in the case of a B1 backcross, we partition the sample into the four ‘ KO ’ groups: $KKOO$, $KKOo$, $KkOO$, $KkOo$. Then, we shuffle only the ‘ MN ’ genotypes ($MMNN$, $MMNn$, $MmNN$, $MmNn$) within each ‘ KO ’ group. Finally, to obtain parameter estimates, we apply the chosen model (CIM or RIM1) to each permuted dataset. When scanning a linkage group for QTL, a new permutation is used with each testing interval. Appendix B.6 gives R program code for implementing this permutation for different breeding designs.

Keeping the ‘ MN ’ genotypes together (rather than separately permuting the M ’s and the N ’s) retains the marker distance between M and N consistent with r_{MN} . As this is a permutation and not resampling with replacement, recombination fractions r_{KM} , r_{MN} , r_{NO} , will be the the same in the permuted datasets as in the original dataset.

Resampling the marker genotypes in this way will allow us to test the appropriate H_0 . If there is a QTL in the interval this will appropriately break the relationship between the observed trait value (which is kept tagged to the individual) and the individual’s ‘ MN ’ marker genotype, which is randomly assigned elsewhere. All other genotypes and cofactors remain tagged to the individual and to the original trait value. This retains possible linkage of any external QTL with the outer intervals $K - M$ and $N - O$, by at least preserving the relationship between y and K , and y and O , and between y and all cofactors.

If the QTL is tightly linked to, say, K , then its effect will still be seen in the right places (the K genotype always stays with the right individual, and that individual’s

score is kept and able to speak to the effect of the QTL near K). But if the QTL is tightly linked to M , but in the interval $K - M$, then the shuffling of the M genotype will break that relationship. Therefore, the expected frequencies of ‘ KL ’ groups and ‘ NR ’ groups may be altered by the permutation. However, RIM1 estimates nuisance parameters in the form of b_L , r_{KL} , b_R and r_{NR} , which do not need to be the equal from replicate to replicate.

Suppose that $\hat{T}_2^{(1)}, \hat{T}_2^{(2)}, \dots, \hat{T}_2^{(N)}$ and $\hat{J}_2^{(1)}, \hat{J}_2^{(2)}, \dots, \hat{J}_2^{(N)}$ are estimates of T_2 and J_2 obtained from N permutations that reflect the null hypothesis. Suppose that \hat{T}_2 and \hat{T}_2 are the corresponding estimates from the original sample. Let

$$\mathcal{J}(a \leq b) = \begin{cases} 1 & \text{if } a \leq b \\ 0 & \text{if } a > b. \end{cases}$$

Then the empirical null-distribution of T_2 is

$$P(\bullet \leq T_2) = \frac{1}{N} \sum_{i=1}^N \mathcal{J}(\hat{T}_2^{(i)} \leq T_2). \quad (5.109)$$

Likewise, the empirical null-distribution of J_2 is

$$P(\bullet \leq J_2) = \frac{1}{N} \sum_{i=1}^N \mathcal{J}(\hat{J}_2^{(i)} \leq J_2). \quad (5.110)$$

The empirical p-values for \hat{T}_2 and \hat{T}_2 may then be calculated as:

$$\text{p-value of } \hat{T}_2 = 1 - P(\bullet \leq \hat{T}_2) \quad (5.111)$$

$$\text{p-value of } \hat{J}_2 = 1 - P(\bullet \leq \hat{J}_2). \quad (5.112)$$

The results of this permutation test are displayed in Chapter 7 and are based on 1000 permutations at each testing interval. The results show that the permutation method proposed here is also very sensitive to the significance level of the test. This sensitivity is more severe when the sample size is small than when it is large.

A key question is how many replicates to create. For their permutation tests, Churchill and Doerge (1994) recommended at least 1000 shuffles to be used for estimating critical values at significance level $\alpha = 0.5$ and as many as 10,000 shuffles for smaller significance levels such as $\alpha = 0.01$. This appears to be a reasonable recommendation. It is clear that for small significance levels, much more than 1000 permutations may be needed to obtain stable estimates of the critical value. This is particularly true when the original sample size is also small.

5.4 Computational Issues

5.4.1 Selecting starting points for the EM Algorithm

Böhning *et al.* (1992) and Seidel *et al.* (2000) have independently shown that, for certain mixture distributions, the parameter estimators obtained from the EM algorithm can depend strongly on the starting strategies and stopping rules used in its implementation. The following quotation, from Lesperance and Lindsay (2001), points out one feature of iterative maximum likelihood procedures that can cause the MLEs generated by these procedures to be dependent on the starting value.

“In a multimodal likelihood, an algorithm tends to go to a root nearest the initial value. Thus it is wise to either search over the space of initial values or to use starting values known to have good properties.”

In our QTL mapping problem, a fixed value of ψ is required to begin the iterative procedure for maximizing the likelihood function. If the vector of mixing parameters, ϕ , is fixed then a unique maximum likelihood estimator for $(\beta, \sigma^2)^T$ exists (and may be calculated using Equations (5.67) and (5.68)). Therefore one only needs to test different mixing parameters in a grid-search for starting values. Potential starting values consist of the test starting-point together with its maximum likelihood

estimators for the variance, QTL effects and cofactor effects. The strategy used in this application consisted of three steps.

1. Choosing a domain of test starting-points.
2. Developing criteria for selecting a point from that domain to form the seed (starting parameter values) for the EM algorithm.
3. Running the EM algorithm from the chosen starting point.

A straight-forward approach to choosing a domain would be to assume Haldane's addition formula for recombination fractions and use a three-dimensional array of points (p_{L2}, p_{Q2}, p_{R2}) , generated by taking evenly spaced points along the interval $(0, 1)$ in each dimension. Due to the structure of the modelled genetic map, some dangers of using such a simple approach were apparent. For example, if $p_{L2} \simeq 0$ and $p_{Q2} \simeq 1$ then loci L , M and Q effectively coincide. This would lead to an ill-

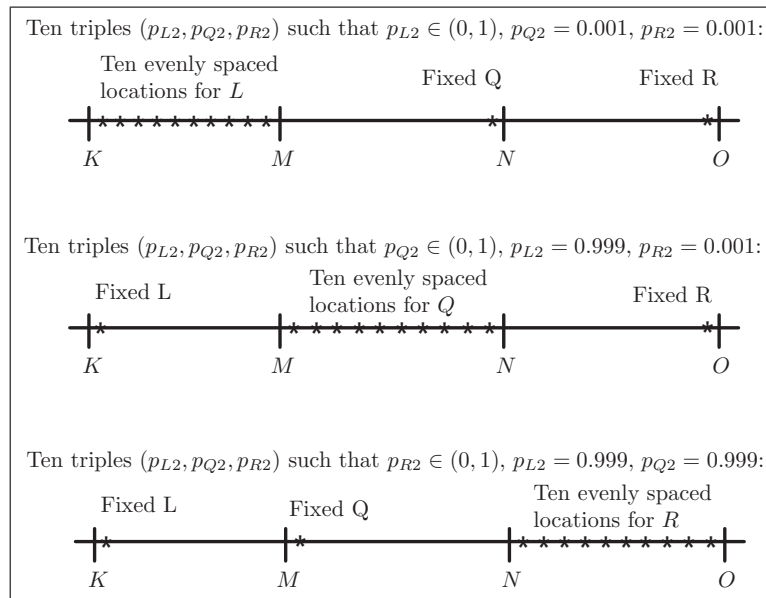


Figure 5.4: Grid of 30 points used as a domain for selecting starting values for the EM Algorithm

conditioned model matrix, and if the corresponding points are used to start the EM algorithm, it is likely that the resulting estimators will be unreliable. This problem was avoided by restricting the grid to ensure that, for all test starting-points, the putative QTL were spaced well away from each other. Figure 5.4 shows the design of the reduced grid. This grid has the added advantage of controlling over-specification in the model and also speeding up the search for starting values. Note that the grid search was only used for selecting starting values. The EM algorithm, as implemented for this work, does not use a grid search to find the maximum likelihood estimator.

Having selected the domain for generating starting values, the next step was to decide upon a criterion for selecting the best starting point from amongst points in the domain. Simulations based on different QTL configurations, showed that the likelihood surface from the grid could be quite uninformative. For some simulated samples the likelihood surface was very flat, for other samples it appeared to be highly multimodal. Only occasionally did it appear to be well behaved. Consequently, the decision was made not to use maximum likelihood alone as the criterion for selecting starting values.

There is precedence in the literature for choosing starting values using criteria other than that of maximum likelihood. Asymptotic likelihood theory (see Lehmann and Casella, 1998, Chapter 6) indicates that, in the case of likelihood estimation in the presence of multiple roots, consistent and efficient estimators may be obtained by taking starting values from a sequence of consistent (but not necessarily efficient) estimators.

Everitt and Hand (1981, pages 47-48) and Lindsay (1995, pages 65-66) discuss some of the strategies for obtaining starting values which have been proposed in the mixture-modelling literature. These include *ad hoc* methods, multiple random starts, graphical techniques, nonparametric likelihood estimation of the latent distribution, method of moments and clustering techniques such as k-means.

For this thesis, a new criterion specially suited to the QTL mapping problem was developed. The new strategy operates by selecting a point that will minimize the environmental variance (σ^2), while maximizing both the variance between the marker groups that we condition on, and the variability between the combined QTL and cofactor classes that occur within each marker class.

Let $\hat{\sigma}^2$ and $\hat{\beta}$ be as given in Equations (5.67) and (5.68). Let ℓ_i be the number of distinct cofactor groups observed within the i^{th} marker group and let n_{ic} be the number of observations belonging to marker i and the c^{th} cofactor group. Also, let \bar{y}_{ick} be the sample mean for any individual belonging to marker group i and having the c^{th} cofactor and k^{th} QTL genotype, and let $n_{ick} \simeq n_{ic}w_{ik}$ be the number of individuals in this category.

Refer to Table 3.1 for the definitions of σ_{error}^2 , σ_i^2 and σ_{total}^2 . Now define:

$$\begin{aligned}\hat{\sigma}_{\text{error}}^2 &= \frac{1}{n}(\mathbf{y} - \mathbf{X}\hat{\beta})^T(\mathbf{y} - \mathbf{X}\hat{\beta}) \\ &= \hat{\sigma}^2 - \frac{1}{n}\hat{\beta}^T \left[(\widetilde{\mathbf{X}^T \mathbf{X}}) - \widetilde{\mathbf{X}}^T \widetilde{\mathbf{X}} \right] \hat{\beta}\end{aligned}\quad (5.113)$$

Assume that the y_{ick} are normally distributed with variance equal to σ_{error}^2 and with mean equal to μ_{ick} . The mean, μ_{ick} , is approximately equal to \bar{y}_{ick} which is estimated by $\hat{\mu}_{ick}$ (where $\hat{\mu}_{ick}$ is the mean of the fitted values in group ick). The variance (σ_i^2) of the trait values within marker group i may be estimated by Equation (5.114) below.

$$\hat{\sigma}_i^2 = \hat{\sigma}_{\text{error}}^2 + \sum_{c=1}^{\ell_i} \sum_{k=1}^t w_{ik} \hat{\mu}_{ick}^2 - \left(\sum_{c=1}^{\ell_i} \sum_{k=1}^t w_{ik} \hat{\mu}_{ick} \right)^2 \quad (5.114)$$

We are also interested in $\text{cov}(\bar{y}_i, \bar{y})$.

$$\begin{aligned}E(\bar{y}_i \bar{y}) &= E\left(\frac{\bar{y}_i}{n} (n_i \bar{y}_i + \sum_{i' \neq i} n_{i'} \bar{y}_{i'}) \right) \\ &= E\left(\frac{n_i}{n} \bar{y}_i^2 + \sum_{i' \neq i} \frac{n_{i'}}{n} \bar{y}_i \bar{y}_{i'} \right) \\ &= \frac{n_i}{n} E(\bar{y}_i^2) + \sum_{i' \neq i} \frac{n_{i'}}{n} E(\bar{y}_i) E(\bar{y}_{i'}) \\ &= \frac{n_i}{n} \left(\frac{\sigma_i^2}{n_i} + E^2(\bar{y}_i) \right) + \sum_{i' \neq i} \frac{n_{i'}}{n} E(\bar{y}_i) E(\bar{y}_{i'})\end{aligned}\quad (5.115)$$

$$\begin{aligned}
\text{cov}(\bar{y}_i, \bar{y}) &= E(\bar{y}_i \bar{y}) - E(\bar{y}_i)E(\bar{y}) \\
&= E(\bar{y}_i \bar{y}) - E(\bar{y}_i) \left(\frac{n_i}{n} E(\bar{y}_i) + \sum_{i' \neq i} \frac{n_{i'}}{n} E(\bar{y}_{i'}) \right) \\
&= E(\bar{y}_i \bar{y}) - \left(\frac{n_i}{n} E^2(\bar{y}_i) + \sum_{i' \neq i} \frac{n_{i'}}{n} E(\bar{y}_i)E(\bar{y}_{i'}) \right) \\
&= \frac{\sigma_i^2}{n} \text{ using the Equation (5.115) above.}
\end{aligned} \tag{5.116}$$

Similar arguments lead to the result

$$\text{cov}(\bar{y}_{ick}, \bar{y}_i) = \frac{\sigma_{\text{error}}^2}{n_i}. \tag{5.117}$$

Define:

$$\begin{aligned}
V_m &= \sum_{i=1}^s \frac{n_i}{n} \text{var}(\bar{y}_i - \bar{y}) \\
&= \sum_{i=1}^s \frac{n_i}{n} (\text{var}(\bar{y}_i) + \text{var}(\bar{y}) - 2 \text{cov}(\bar{y}_i, \bar{y})) \\
&= \sum_{i=1}^s \frac{n_i}{n} \left(\frac{\sigma_i^2}{n_i} + \frac{\sigma_{\text{total}}^2}{n} - \frac{2\sigma_i^2}{n} \right).
\end{aligned} \tag{5.118}$$

Substituting $\hat{\sigma}^2$ for σ_{total}^2 and $\hat{\sigma}_i^2$ for σ_i^2 we have the approximation

$$\hat{V}_m = \sum_{i=1}^s \frac{1}{n^2} (n_i \hat{\sigma}^2 + (n - 2n_i) \hat{\sigma}_i^2). \tag{5.119}$$

If there are QTL interior to the central testing interval then V_m should be significantly different from zero.

Now define:

$$\begin{aligned}
V_{qc} &= \sum_{i=1}^s \frac{n_i}{n} \sum_{c=1}^{\ell_i} \sum_{k=1}^t \frac{n_{ick}}{n_i} \text{var}(\bar{y}_{ick} - \bar{y}_i) \\
&= \sum_{i=1}^s \sum_{c=1}^{\ell_i} \sum_{k=1}^t \frac{n_{ick}}{n} (\text{var}(\bar{y}_{ick}) + \text{var}(\bar{y}_i) - 2 \text{cov}(\bar{y}_{ick}, \bar{y}_i)) \\
&= \sum_{i=1}^s \sum_{c=1}^{\ell_i} \sum_{k=1}^t \frac{n_{ick}}{n} \left(\frac{\sigma_{\text{error}}^2}{n_{ick}} + \frac{\sigma_i^2}{n_i} - \frac{2\sigma_{\text{error}}^2}{n_i} \right).
\end{aligned} \tag{5.120}$$

$$\widehat{V}_{qc} = \sum_{i=1}^s \sum_{c=1}^{\ell_i} \sum_{k=1}^t \frac{1}{n_i n} (n_{ic} \widehat{w}_{ik} \widehat{\sigma}_i^2 + (n_i - 2n_{ic} \widehat{w}_{ik}) \widehat{\sigma}_{\text{error}}^2). \quad (5.121)$$

If there are QTL adjacent to the testing interval or QTL located further away and associated with any cofactor, then V_{qc} should be significantly different from zero. If the trait is genetically determined, then $\widehat{V}_m + \widehat{V}_{qc}$ should be much larger than the error variance.

$$\text{Variance Ratio} = \frac{\widehat{\sigma}_{\text{error}}^2}{\widehat{V}_m + \widehat{V}_{qc}} \quad (5.122)$$

The starting value for $\boldsymbol{\psi}$ is taken to be the grid-point (and associated MLEs $\widehat{\boldsymbol{\beta}}, \widehat{\sigma}^2$) which minimizes the variance ratio given in Equation (5.122). Simulations indicated that this variance-ratio-minimization (VRM) criterion was quite good at correctly identifying QTL genotype-clusters within the data because it tended to select starting values that were close to the true values.

5.4.2 Stopping rules

The standard lack-of-progress stopping criterion was used to terminate the EM-algorithm. This criterion terminates the algorithm if changes in the likelihood are smaller than a chosen tolerance value. Let \mathcal{L}_{i-1} and \mathcal{L}_i denote values of the log-likelihood for two consecutive steps of the EM-algorithm. Then the algorithm lack-of-progress stopping criterion is

$$\text{stop if } \mathcal{L}_i - \mathcal{L}_{i-1} < \text{tolerance}.$$

This simple stopping rule sometimes attracts criticism because it is possible for a slow-converging algorithm to take small steps in the likelihood and still be far away from a local maximum. Several alternative stopping rules have been proposed in the literature (see Lindsay, 1995, page 64).

In this thesis, a new stopping rule was also developed and tested. If the mixing parameters are fixed then the MLE is unique. This implies that if the category identities are not changing on successive iterations, then continuing with iterations will not improve the likelihood. Our new stopping rule checks whether successive iterations are changing the category identities of individuals. The stopping rule is

$$\text{stop if } \frac{1}{nt} \mathbf{1}_n^T [\text{abs}(\mathbf{Z}_{next} - \mathbf{Z}_{current})] \mathbf{1}_n < \text{tolerance}$$

where $\mathbf{1}_n$ is the summing vector of order n , $\mathbf{Z}_{current}$ and \mathbf{Z}_{next} are matrices of category identities obtained from consecutive E-steps, n is the number of individuals and t is the number of mixing components. The absolute value of the resultant of the matrix-subtraction, denoted by $\text{abs}(\mathbf{Z}_{next} - \mathbf{Z}_{current})$, operates in an element-wise manner.

Simulations showed that this new criterion was better at preventing premature termination of the algorithm than the lack-of-progress criterion. However, when the tolerance limit for the lack-of-progress criterion was very small (tolerance = 10^{-6}) the two rules seemed to agree. Therefore, because of its simplicity, the standard lack-of-progress stopping rule was preferred.

5.4.3 Adjustments to RIM1

The RIM1 model was defined to condition on the genotypes at four markers. When testing the first or last interval in a linkage group, it is desirable to tweak the model so that it conditions on three markers instead of four.

When testing the first interval in a linkage group, the markers M , N and O are available but marker K is unavailable. We define

$$p_L = P(LL|MM) = 1 - r_{LM}$$

which implies that $P(L\ell|MM) = (1 - p_L)$. Next, we substitute p_L for p_{L1} and $(1 - p_L)$ for p_{L2} when calculating the mixing proportions. Instead of the parameters (p_{L2}, p_{Q2}, p_{R2}) , we have (p_L, p_{Q2}, p_{R2}) .

Similarly, if the dataset is such that the testing interval is the last interval on the right, then marker O is not available. We define

$$p_R = P(RR|NN) = 1 - r_{NO}$$

and substitute p_R for p_{R1} and $(1-p_R)$ for p_{R2} when calculating the mixing proportions. Instead of the parameters (p_{L2}, p_{Q2}, p_{R2}) , we have (p_{L2}, p_{Q2}, p_R) . Conditioning on fewer than three markers is not permitted for RIM1 and conditioning on three markers is only permitted when the testing interval is the first or last interval in a linkage group.

5.4.4 Reduced Models for fitting fewer than three QTL

The Model RIM1 has seven analogous reduced models of which the CIM model is one. The reduced models are determined by removing some putative QTL from RIM1. We have the following models.

‘LQR’=RIM1	‘LR’
‘LQ’	‘L’
‘QR’	‘R’
‘Q’=CIM	‘N’=No QTL

The reduced models are not simply nested within RIM1 because they condition on different numbers of markers than RIM1. For example, the models CIM, ‘L’ and ‘R’ all fit one QTL and need to condition on two flanking markers in order to exploit the properties of interval mapping. Likewise, the models (‘LR’, ‘LQ’ and ‘QR’) which fit two QTL must condition on three flanking makers. Still, the form of the likelihood (and information matrix) is the same for all of the models that contain QTL. However, the dimensions and contents of \mathbf{C} , \mathbf{Z} , $\boldsymbol{\psi}$ and \mathbf{W} differ between models. This makes it easy to write modular program code.

We do not need to write new code to implement each of the seven models. Instead we write modules to calculate the matrices \mathbf{C} , \mathbf{Z} , $\boldsymbol{\psi}$, \mathbf{W} and to calculate the first two derivatives of the natural logarithm of \mathbf{W} with respect to $\boldsymbol{\phi}$, where the calculation depends on which model is being used. Then we feed these objects into a single module that performs the EM maximization, and into a single module that calculates the information matrix (see Appendix B). The model containing no QTL may be implemented as a simple marker-regression. If we are concerned about possible over-specification in RIM1, popular model-selection techniques could be used to choose between these eight models. For example, Akaike's information criterion could be used for this purpose (Akaike, 1974).

5.4.5 The possibility of a singular information matrix

The Fisher information matrix is always positive semi-definite but it is not always positive definite. It is possible to obtain a singular (or nearly singular) Fisher information matrix. This occurrence is indicative of an ill-conditioned system. In a mixture model, an ill conditioned model matrix may occur when there is collinearity in observed data or when certain mixing parameters lie on the boundary of the parameter space.

Silvey (1959) and Breusch (1986) suggest that, in special cases, a generalized inverse could be used to construct hypothesis tests when the information matrix is singular. Rotnitzky *et al.* (2000) and Prescott *et al.* (2002) also developed hypothesis tests for specific models involving singular information matrices.

It is useful to examine how often our proposed model tends to generate a singular information matrix. In our implementation, whenever a singular information matrix is encountered, a warning is returned and the covariance matrix is estimated by the generalised inverse of the information matrix. Applying RIM1 to 20 intervals for 200 simulated backcross samples required 4000 calculations of the information

matrix. None of these 4000 calculations produced a singular information matrix. Likewise, applying the RIM to a real backcross sample and to a real F2 sample did not produce a singular information matrix in either case. However, when a non-parametric bootstrap was applied to one of the simulated data-sets, 11 out of 1000 bootstrap samples yielded a singular information matrix. Similarly, when the real backcross data was bootstrapped, 70 out of 1000 bootstrap samples yielded a singular information matrix. When the analyses were repeated using the CIM model, none of the original data-sets yielded a singular information matrix but similar numbers of bootstrap samples yielded singular information matrices. More details about these simulated and real datasets are given in Chapter 7 and Chapter 8 respectively.

The results suggest that because the EM-algorithm never permits the parameter to fall exactly on the boundary, it is very rare that lack of identifiability will cause our system to produce a singular information matrix. When a singular information matrix occurs in this framework it is more likely due to collinearity in the observed data. The bootstrap technique is based on re-sampling with replacement. Therefore, rare marker groups that are observed in the original sample are likely to be omitted in a bootstrap sample. This can increase the risk of introducing collinearity in our data.

5.4.6 Programming environment

This thesis is not concerned with creating new methods for simulating genetic data and breeding designs. Therefore, all simulated samples were generated using the QTL Cartographer (Basten *et al.*, 1994, 2001) software.

The R language and environment (R Development Core Team (2006)) was chosen for this project because it offers powerful tools for statistical analysis and programming. The flexible indexing and manipulation features associated with its matrix and list objects made R particularly suited for our data analysis.

Most of the data analysis was carried out using bespoke programs written in the R programming language. For the purposes of comparison, Composite Interval Mapping was also carried out using the QTL Cartographer software. All R program code to implement the methodology proposed in this thesis was written exclusively by the author. The core segments of these programs are included in Appendix B to illustrate that proposed methods are practical to implement.

Chapter 6

Information Matrix Derivations

This chapter gives detailed mathematical proofs for the information matrix formulae presented in Equations (5.84) to (5.94) and in Equation (5.97) from the previous chapter. Although the algebraic manipulations presented in this chapter are relatively simple, they are quite tedious. We include these technical details as a Chapter rather than an Appendix because they represent a significant part of the novel contribution of this thesis. Without these mathematical proofs, there would be no evidence the proposed formulae are based on exact derivations.

The proofs do not involve any approximations, and so application of the proposed information matrix formulae does not require any extra assumptions on top of those assumptions needed for asymptotic maximum likelihood theory to hold. Therefore, one can expect that in any situation where classical asymptotic maximum likelihood theory applies, the proposed formulae will give good estimators of the standard errors of the MLEs.

Readers who are not interested in the details of these mathematical proofs may proceed directly to Chapter 7, where we apply the methods that are described in Chapter 5 to some simulated data.

6.1 The Complete-Data Conditional Information

In this section we partition the upper triangle of the complete-data (conditional) information matrix,

$$\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}) = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y}, \mathbf{Z})] = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\boldsymbol{\psi}\boldsymbol{\psi}}],$$

into ten blocks and derive formulae for evaluating the blocks.

The score functions of the complete-data likelihood are

$$\begin{aligned} \mathcal{U}_{\mathbf{b}} &= -\frac{1}{\sigma^2}(\mathbf{X}_1^T \mathbf{X}_1 \mathbf{b} - \mathbf{X}_1^T \mathbf{y} + \mathbf{X}_1^T \mathbf{X}_2 \mathbf{b}^*) \\ &= -\frac{1}{\sigma^2}(\mathbf{X}_1^T \mathbf{X}_1 \mathbf{b} - \mathbf{X}_1^T \ddot{\mathbf{y}}), \\ \mathcal{U}_{\mathbf{b}^*} &= -\frac{1}{\sigma^2}(\mathbf{X}_2^T \mathbf{X}_2 \mathbf{b}^* - \mathbf{X}_2^T \mathbf{y} + \mathbf{X}_2^T \mathbf{X}_1 \mathbf{b}) \\ &= -\frac{1}{\sigma^2}(\mathbf{X}_2^T \mathbf{X}_1 \mathbf{b} - \mathbf{X}_2^T \ddot{\mathbf{y}}), \\ \mathcal{U}_{(\sigma^2)} &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\mathbf{y} - \mathbf{X}_1 \mathbf{b} - \mathbf{X}_2 \mathbf{b}^*)^T (\mathbf{y} - \mathbf{X}_1 \mathbf{b} - \mathbf{X}_2 \mathbf{b}^*) \\ &= -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\ddot{\mathbf{y}} - \mathbf{X}_1 \mathbf{b})^T (\ddot{\mathbf{y}} - \mathbf{X}_1 \mathbf{b}), \\ \mathcal{U}_{\boldsymbol{\phi}} &= \sum_{i=1}^s \left(\frac{\partial}{\partial \boldsymbol{\phi}} \mathbf{h}_i^T(\boldsymbol{\phi}) \right) \mathbf{Z}_i^T \mathbf{1}_{n_i} = \sum_{i=1}^s \left(\frac{\partial}{\partial \boldsymbol{\phi}} \mathbf{h}_i^T(\boldsymbol{\phi}) \right) \mathbf{Z}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i}. \end{aligned}$$

Their conditional expectations are therefore:

$$E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}] = -\frac{1}{\sigma^2} \left(\mathbf{C}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \mathbf{C} \mathbf{b} - \mathbf{C}^T \tilde{\mathbf{Z}}^T \mathbf{y} \right) \quad (6.1)$$

$$E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}] = -\frac{1}{\sigma^2} (\mathbf{X}_2^T \tilde{\mathbf{Z}} \mathbf{C} \mathbf{b} - \mathbf{X}_2^T \ddot{\mathbf{y}}) \quad (6.2)$$

$$E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}] = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} \left(\mathbf{y}^T \mathbf{y} - 2\mathbf{y}^T \tilde{\mathbf{Z}} \mathbf{C} \mathbf{b} + \mathbf{b}^T \mathbf{C}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \mathbf{C} \mathbf{b} \right) \quad (6.3)$$

$$E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\phi}}] = \sum_{i=1}^s \left(\frac{\partial}{\partial \boldsymbol{\phi}} \mathbf{h}_i^T(\boldsymbol{\phi}) \right) \tilde{\mathbf{Z}}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i}. \quad (6.4)$$

These conditional expectations (Equations (6.1) to (6.4)) specify the components of $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}}]$ which are not relevant for calculating $\mathcal{I}_c(\boldsymbol{\psi}; \mathbf{y})$. However, these components are presented here because they will be used later, in the calculation of $\mathcal{I}_m(\boldsymbol{\psi}; \mathbf{y})$, the conditional missing information.

The second partial derivatives of the complete-data log-likelihood are:

$$\begin{aligned}\mathcal{U}_{\mathbf{b}\mathbf{b}} &= -\frac{1}{\sigma^2} \mathbf{X}_1^T \mathbf{X}_1 \\ \mathcal{U}_{\mathbf{b}^*\mathbf{b}^*} &= -\frac{1}{\sigma^2} \mathbf{X}_2^T \mathbf{X}_2 \\ \mathcal{U}_{(\sigma^2)(\sigma^2)} &= \frac{n}{2\sigma^4} - \frac{1}{\sigma^6} (\ddot{\mathbf{y}} - \mathbf{X}_1 \mathbf{b})^T (\ddot{\mathbf{y}} - \mathbf{X}_1 \mathbf{b}) \\ \mathcal{U}_{\phi\phi} &= \sum_{i=1}^s \frac{\partial}{\partial \phi} \left[\frac{\partial}{\partial \phi^T} (\mathbf{h}_i^T(\phi)) \mathbf{Z}_i^T \mathbf{1}_{n_i} \right]\end{aligned}$$

$$\begin{aligned}\mathcal{U}_{\mathbf{b}\mathbf{b}^*} &= -\frac{1}{\sigma^2} \mathbf{X}_1^T \mathbf{X}_2 \\ \mathcal{U}_{\mathbf{b}(\sigma^2)} &= \frac{1}{\sigma^4} (\mathbf{X}_1^T \mathbf{X}_1 \mathbf{b} - \mathbf{X}_1^T \ddot{\mathbf{y}}) \\ \mathcal{U}_{\mathbf{b}^*(\sigma^2)} &= \frac{1}{\sigma^4} (\mathbf{X}_2^T \mathbf{X}_1 \mathbf{b} - \mathbf{X}_2^T \ddot{\mathbf{y}}) \\ \mathcal{U}_{\mathbf{b}\phi} &= \mathbf{0} \\ \mathcal{U}_{\mathbf{b}^*\phi} &= \mathbf{0} \\ \mathcal{U}_{(\sigma^2)\phi} &= \mathbf{0}\end{aligned}$$

Evaluating $E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\psi\psi}]$ simply involves taking the conditional expectations of (-1) times the second partial derivatives given above. Therefore the ten blocks of that form the upper triangle of $E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\psi\psi}]$ (the complete-data information matrix conditioned on the observed data) are as given below.

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}\mathbf{b}}] = \frac{1}{\sigma^2} \mathbf{C}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \mathbf{C} \quad (6.5)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}^*\mathbf{b}^*}] = \frac{1}{\sigma^2} \mathbf{X}_2^T \mathbf{X}_2 \quad (6.6)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{(\sigma^2)(\sigma^2)}] = -\frac{n}{2\sigma^4} + \frac{1}{\sigma^6} \left(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}} - 2\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} \mathbf{C} \mathbf{b} + \mathbf{b}^T \mathbf{C}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \mathbf{C} \mathbf{b} \right) \quad (6.7)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\phi\phi}] = -\sum_{i=1}^s \left(\frac{\partial^2}{\partial \phi \partial \phi^T} \mathbf{h}_i^T(\phi) \right) \tilde{\mathbf{Z}}_i^T \mathbf{1}_{n_i} \quad (6.8)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}\mathbf{b}^*}] = \frac{1}{\sigma^2} \mathbf{C}^T \tilde{\mathbf{Z}}^T \mathbf{X}_2 \quad (6.9)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}(\sigma^2)}] = -\frac{1}{\sigma^4} \left(\mathbf{C}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \mathbf{C} \mathbf{b} - \mathbf{C}^T \tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \right) \quad (6.10)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}^*(\sigma^2)}] = -\frac{1}{\sigma^4} (\mathbf{X}_2^T \tilde{\mathbf{Z}} \mathbf{C} \mathbf{b} - \mathbf{X}_2^T \ddot{\mathbf{y}}) \quad (6.11)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}\phi}] = \mathbf{0} \quad (6.12)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}^*\phi}] = \mathbf{0} \quad (6.13)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{(\sigma^2)\phi}] = \mathbf{0} \quad (6.14)$$

6.2 Notation and Useful Matrix Identities

6.2.1 Notation

In this section and in all other sections of Chapter 6 we use the following notation.

- (1) $\boldsymbol{\mu} = \mathbf{C} \mathbf{b}$ for the column vector of means.
- (2) $\text{diag}(\mathbf{v}^T)$ to denote the diagonal matrix whose i^{th} diagonal element is given by the i^{th} element of \mathbf{v}^T , where \mathbf{v}^T is a row vector.
- (3) $\mathbf{A}_{k\bullet}$ to denote the k^{th} row of a matrix \mathbf{A} .
- (4) $\mathbf{A}_{\bullet k}$ to denote the k^{th} column of a matrix \mathbf{A} .
- (5) s to denote the number of (marker) groupings on which we condition. For RIM1, $s = 16$ for a backcross design, while $s = 81$ for a $F2$ design.
- (6) t to denote the number of mixture components (QTL groupings). For RIM1, $t = 8$ for a backcross design, while $t = 27$ for a $F2$ design.
- (7) $\tilde{\mathbf{D}}_k = \text{diag}((\tilde{\mathbf{Z}}_{\bullet k})^T)$, where $\tilde{\mathbf{Z}}$ is the matrix of imputed category identities that is defined in Equation (5.50). Note that $\tilde{\mathbf{D}}_k$ is an $n \times n$ diagonal matrix.

- (8) In order to allow the separating of marker genotypes, as and when necessary, we may use double indexing (ij) . A reference to row (ij) will correspond to a row that stores data for the j^{th} individual in group i . Likewise, we may use the double index $(i'j')$ to refer to a single column that stores data for individual j' in group i' . We may also use a pair of double subscripts when referring to a cell of a matrix \mathbf{A} , such as the cell $\mathbf{A}_{(ij)(i'j')}$, which represents a scalar.
- (9) \mathbf{I}_t to denote the $t \times t$ identity matrix.
- (10) \mathbf{I}_n to denote the $n \times n$ identity matrix, and $[\mathbf{I}_n]_{\bullet(ij)}$ to denote its $(ij)^{\text{th}}$ column.

6.2.2 General Matrix Identities

This section defines five general matrix identities that will be used later. For these definitions, let:

- (1) \mathbf{u} be a $t \times 1$ vector;
- (2) \mathbf{v} and \mathbf{a} be $n \times 1$ vectors, with the k^{th} element equal to v_k and a_k respectively;
- (3) \mathbf{A} be an $n \times t$ matrix;
- (4) $\text{diag}(\mathbf{v}^T)$ be an $n \times n$ diagonal matrix with its i^{th} diagonal entry equal to v_i ;
- (5) $\mathbf{1}_n$ be an $n \times 1$ vector of ones.

Then, the five identities displayed in Equations (6.15) to (6.19), below, hold true.

$$\mathbf{A}_{k\bullet}\mathbf{u} = [\mathbf{A}\mathbf{u}]_{k\bullet} \quad (6.15)$$

$$\mathbf{v}^T \mathbf{A}_{\bullet k} = [\mathbf{v}^T \mathbf{A}]_{\bullet k} \quad (6.16)$$

$$\mathbf{1}_n^T \text{diag}(\mathbf{A}_{\bullet k}) = \mathbf{A}_{\bullet k} \quad (6.17)$$

$$\mathbf{A}_{\bullet k} \mathbf{u}_k = [\mathbf{A} \text{diag}(\mathbf{u})]_{\bullet k} \quad (6.18)$$

$$v_k a_k = [\text{diag}(\mathbf{v}) \mathbf{a}]_{k\bullet} = [\mathbf{a}^T \text{diag}(\mathbf{v})]_{\bullet k} = [\text{diag}(\mathbf{a}) \mathbf{v}]_{k\bullet} = [\mathbf{v}^T \text{diag}(\mathbf{a})]_{\bullet k} \quad (6.19)$$

6.2.3 Matrix Identities that are Specific to our Problem

This section lists some incidental results that are useful for simplifying the calculation of $\mathcal{I}_m(\boldsymbol{\psi}; \mathbf{y})$, the conditional missing information. These results are displayed in Equations (6.20) to (6.48) below. The definitions of all notation used here may be found in Chapter 5 and in Section 6.2.1. Note that these incidental results are just specific applications of the matrix identities given in Equations (6.15) to (6.19) above. These results are collected together here because this arrangement enables easy referencing. The usefulness of each formula will become more apparent later, when we encounter them in the calculation of $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}} \mathcal{U}_{\boldsymbol{\psi}}^T]$ (see Propositions 6.4.1 to 6.4.10).

$$\mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k} \mu_k = [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})]_{k\bullet} \boldsymbol{\mu} = [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu}]_{k\bullet} \quad (6.20)$$

$$(\tilde{\mathbf{Z}}_{\bullet k'})^T \mathbf{1}_n \mu_{k'} = [\boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})]_{\bullet k'} \quad (6.21)$$

$$\mathbf{1}_n^T \tilde{\mathbf{D}}_k \mu_k = (\tilde{\mathbf{Z}}_{\bullet k})^T \mu_k = \mu_k (\tilde{\mathbf{Z}}_{\bullet k})^T = [\text{diag}(\boldsymbol{\mu}^T)]_{k\bullet} \tilde{\mathbf{Z}}^T = [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T]_{k\bullet} \quad (6.22)$$

$$\mu_{k'} \tilde{\mathbf{D}}_{k'} \mathbf{1}_n = [\tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T)]_{\bullet k'} \quad (6.23)$$

$$\begin{aligned} \mathbf{1}_n^T \tilde{\mathbf{D}}_k \mathbf{1}_n \mu_{k'}^2 &= \mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k} \mu_{k'}^2 \\ &= [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})]_{k\bullet} \mu_{k'}^2 \\ &= [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})]_{k\bullet} [\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T)]_{\bullet k'} \end{aligned} \quad (6.24)$$

$$(\tilde{\mathbf{Z}}_{\bullet k'})^T \ddot{\mathbf{y}} = [\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}}]_{k'\bullet} = [\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}]_{\bullet k'} \quad (6.25)$$

$$\tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}} = \text{diag}((\tilde{\mathbf{Z}}_{\bullet k'})^T) \ddot{\mathbf{y}} = \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}_{\bullet k'} = [\text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{\bullet k'} \quad (6.26)$$

$$\begin{aligned}
\mathbf{1}_n^T \tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}} \mu_k &= (\tilde{\mathbf{Z}}_{\bullet k'})^T \ddot{\mathbf{y}} \mu_k \\
&= [\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}]_{\bullet k'} \mu_k \\
&= \mu_k [\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}]_{\bullet k'} \\
&= [\text{diag}(\boldsymbol{\mu}^T)]_{k\bullet} [\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}})]_{\bullet k'}
\end{aligned} \tag{6.27}$$

$$\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}_{\bullet k} = [\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}]_{\bullet k} = [\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}}]_{k\bullet} \tag{6.28}$$

$$\ddot{\mathbf{y}}^T \tilde{\mathbf{D}}_k = \ddot{\mathbf{y}}^T \text{diag}((\tilde{\mathbf{Z}}_{\bullet k'})^T) = (\tilde{\mathbf{Z}}_{\bullet k})^T \text{diag}(\ddot{\mathbf{y}}^T) = [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T)]_{k\bullet} \tag{6.29}$$

$$\begin{aligned}
\ddot{\mathbf{y}}^T \tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}} &= \ddot{\mathbf{y}}^T [\text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{\bullet k'} \\
&= [\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{\bullet k'} \\
&= \left[\text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) \right]_{k'k'}
\end{aligned} \tag{6.30}$$

$$\mathbf{1}_{n_i}^T \Delta_i \tilde{\mathbf{Z}}_{\bullet k} = [\mathbf{1}_{n_i}^T \Delta_i \tilde{\mathbf{Z}}]_{\bullet k} = [\tilde{\mathbf{Z}}^T \Delta_i^T \mathbf{1}_{n_i}]_{k\bullet} = [\tilde{\mathbf{Z}}_i^T \mathbf{1}_{n_i}]_{k\bullet} \tag{6.31}$$

$$(\tilde{\mathbf{Z}}_{\bullet k'})^T \Delta_{i'}^T \mathbf{1}_{n_{i'}} = [\mathbf{1}_{n_{i'}}^T \tilde{\mathbf{Z}}_{i'}]_{\bullet k'} \tag{6.32}$$

$$\mathbf{1}_{n_i}^T \Delta_i \tilde{\mathbf{D}}_k = (\tilde{\mathbf{Z}}_{\bullet k})^T \Delta_i^T \Delta_i = [\tilde{\mathbf{Z}}^T \Delta_i^T \Delta_i]_{k\bullet} = [\tilde{\mathbf{Z}}_i^T \Delta_i]_{k\bullet} \tag{6.33}$$

$$\tilde{\mathbf{D}}_{k'} \Delta_{i'}^T \mathbf{1}_{n_{i'}} = [\Delta_{i'}^T \tilde{\mathbf{Z}}_{i'}]_{\bullet k'} \tag{6.34}$$

$$\begin{aligned}
\mathbf{1}_{n_i}^T \Delta_i \tilde{\mathbf{D}}_{k'} \Delta_{i'}^T \mathbf{1}_{n_{i'}} &= [\tilde{\mathbf{Z}}_i^T \Delta_i]_{k'\bullet} \Delta_{i'}^T \mathbf{1}_{n_{i'}} \\
&= [\tilde{\mathbf{Z}}_i^T \Delta_i \Delta_{i'}^T \mathbf{1}_{n_{i'}}]_{k'\bullet} \\
&= [\mathbf{1}_{n_{i'}}^T \Delta_{i'} \Delta_i^T \tilde{\mathbf{Z}}_i]_{\bullet k'} \\
&= [\text{diag}(\mathbf{1}_{n_{i'}}^T \Delta_{i'} \Delta_i^T \tilde{\mathbf{Z}}_i)]_{k'k'}
\end{aligned} \tag{6.35}$$

$$\begin{aligned}
\mathbf{1}_n^T \tilde{\mathbf{D}}_k \mu_k \Delta_i^T \mathbf{1}_{n_i} &= [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T]_{k\bullet} \Delta_i^T \mathbf{1}_{n_i} \\
&= [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \Delta_i^T \mathbf{1}_{n_i}]_{k\bullet} \\
&= [\mathbf{1}_{n_i}^T \Delta_i \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T)]_{k\bullet} \\
&= [\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)]_{k\bullet} \\
&= \left[\text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) \right]_{kk}
\end{aligned} \tag{6.36}$$

$$\begin{aligned}
\ddot{\mathbf{y}}^T \tilde{\mathbf{D}}_{k'} \Delta_i^T \mathbf{1}_{n_i} &= \ddot{\mathbf{y}}^T [\Delta_i^T \tilde{\mathbf{Z}}_i]_{\bullet k'} \\
&= [\ddot{\mathbf{y}}^T \Delta_i^T \tilde{\mathbf{Z}}_i]_{\bullet k'} \\
&= \left[\text{diag}(\ddot{\mathbf{y}}^T \Delta_i^T \tilde{\mathbf{Z}}_i) \right]_{k'k'}
\end{aligned} \tag{6.37}$$

$$\mu_k \mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k} = \mu_k \tilde{m}_k = \mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k} \mu_k = [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})]_{k\bullet} \boldsymbol{\mu} = [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu}]_{k\bullet} \tag{6.38}$$

$$\tilde{\mathbf{Z}}_{(ij)\bullet} \boldsymbol{\mu} = [\tilde{\mathbf{Z}} \boldsymbol{\mu}]_{(ij)\bullet} = [\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T]_{\bullet(ij)} \tag{6.39}$$

$$\mu_k \mathbf{1}_n^T [\mathbf{I}_n]_{\bullet(ij)} = \mu_k = \boldsymbol{\mu}_{k\bullet} \tag{6.40}$$

$$[\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{k\bullet} \boldsymbol{\mu} = \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \mu_k \tag{6.41}$$

$$\mu_k \mathbf{1}_n^T \tau_{ik}(y_{ij}; \boldsymbol{\psi}) [\mathbf{I}_n]_{\bullet(ij)} = \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \mu_k \mathbf{1}_n^T [\mathbf{I}_n]_{\bullet(ij)},$$

since $\tau_{ik}(y_{ij}; \boldsymbol{\psi})$ is a scalar

$$= \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \mu_k \tag{6.42}$$

$$\ddot{\mathbf{y}}^T [\mathbf{I}_n]_{\bullet(ij)} = \ddot{y}_{ij} \quad (6.43)$$

$$\begin{aligned} \ddot{\mathbf{y}}^T \tau_{ik}(y_{ij}; \boldsymbol{\psi}) [\mathbf{I}_n]_{\bullet(ij)} &= \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \ddot{\mathbf{y}}^T [\mathbf{I}_n]_{\bullet(ij)}, \\ &\text{since } \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \text{ is a scalar} \\ &= \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \ddot{y}_{ij} \end{aligned} \quad (6.44)$$

$$\boldsymbol{\mu}^T (\tilde{\mathbf{Z}}_{(ij)\bullet})^T = \boldsymbol{\mu}^T \tilde{\mathbf{Z}}_{\bullet(ij)}^T = [\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T]_{\bullet(ij)} = [\tilde{\mathbf{Z}}\boldsymbol{\mu}]_{(ij)\bullet}. \quad (6.45)$$

$$\begin{aligned} \boldsymbol{\mu}^T \text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet}) \boldsymbol{\mu} &= \boldsymbol{\mu}^T [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T]_{\bullet(ij)} \\ &= [\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T]_{\bullet(ij)} \end{aligned} \quad (6.46)$$

$$\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i [\mathbf{I}_n]_{\bullet(i'j')} = [\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{I}_n]_{\bullet(i'j')} = [\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i]_{\bullet(i'j')} \quad (6.47)$$

$$\begin{aligned} \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \tau_{i'k}(y_{i'j'}; \boldsymbol{\psi}) [\mathbf{I}_n]_{\bullet(i'j')} &= \tau_{i'k}(y_{i'j'}; \boldsymbol{\psi}) \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i [\mathbf{I}_n]_{\bullet(i'j')}, \\ &\text{since } \tau_{i'k}(y_{i'j'}; \boldsymbol{\psi}) \text{ is a scalar} \\ &= \tau_{i'k}(y_{i'j'}; \boldsymbol{\psi}) [\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{I}_n]_{\bullet(i'j')} \\ &= \tau_{i'k}(y_{i'j'}; \boldsymbol{\psi}) [\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i]_{\bullet(i'j')}, \text{ by definition of } \mathbf{I}_n \\ &= [\tilde{\mathbf{Z}}^T \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i)]_{k(i'j')} \end{aligned} \quad (6.48)$$

6.3 Conditional Expectations of Products of the Estimated Category Identities

Like the previous section, this section also presents intermediate results that are used to simplify the calculation of the conditional missing information. Table 6.1, below, lists the calculations that are dealt with in this section.

Table 6.1: Selected conditional expectations involving products of estimated category identities

Conditional expectation	Proposition that reveals a formula for its calculation
$E_{\mathbf{Z} \mathbf{y};\psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]$	Proposition 6.3.1
$E_{\mathbf{Z} \mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}]$	Proposition 6.3.2
$E_{\mathbf{Z} \mathbf{y};\psi}[(\mathbf{Z}_{(ij)\bullet})^T\mathbf{Z}_{(i'j')\bullet}]$	Proposition 6.3.3

Proposition 6.3.1.

$$E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T] = \tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T - \tilde{\mathbf{D}}_k\tilde{\mathbf{D}}_{k'} + \delta_{kk'}\tilde{\mathbf{D}}_{k'}$$

where $\delta_{kk'}$ is the Kronecker delta, which has value one if $k = k'$ and zero otherwise, and where $\tilde{\mathbf{D}}_k = \text{diag}((\tilde{\mathbf{Z}}_{\bullet k})^T)$ and $\tilde{\mathbf{D}}_{k'} = \text{diag}((\tilde{\mathbf{Z}}_{\bullet k'})^T)$.

Proof of Proposition 6.3.1. We begin by showing that the matrices $\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T$ and $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]$ differ only in their diagonal elements. Then, we show that whenever k is not equal to k' , the diagonal elements of $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]$ are equal to zero, but if $k = k'$ the diagonal elements of $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]$ are given by the elements of the row vector $(\tilde{\mathbf{Z}}_{\bullet k})^T$.

The matrix $\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T$ is $n \times n$ and the element in row (ij) and column $(i'j')$ is given by

$$[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]_{(ij)(i'j')} = z_{ijk}z_{i'j'k'}.$$

However, by definition of \mathbf{Z} ,

$$z_{ijk}z_{i'j'k'} = \begin{cases} 0 & \text{if } (ij) = (i'j') \text{ and } k \neq k' \\ z_{ijk} & \text{if } (ij) = (i'j') \text{ and } k = k' \\ z_{ijk}z_{i'j'k'} & \text{if } (ij) \neq (i'j') \text{ for all } k \text{ and } k' \end{cases}$$

Therefore,

$$E_{\mathbf{Z}|\mathbf{y};\boldsymbol{\psi}}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]_{(ij)(i'j')} = \begin{cases} 0, & \text{if } (ij) = (i'j') \text{ and } k \neq k' \\ \tau_{ik}(y_{ij}; \boldsymbol{\psi}), & \text{if } (ij) = (i'j') \text{ and } k = k' \\ \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{i'k'}(y_{i'j'}; \boldsymbol{\psi}), & \text{if } (ij) \neq (i'j'), \text{ (by} \\ & \text{independence of individuals).} \end{cases}$$

The matrix $\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T$ is also $n \times n$ and the element in row (ij) and column $(i'j')$ is given by

$$[\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T]_{(ij)(i'j')} = \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{i'k'}(y_{i'j'}; \boldsymbol{\psi}).$$

Now let $\tilde{\mathbf{D}}_k = \text{diag}((\tilde{\mathbf{Z}}_{\bullet k})^T)$ and $\tilde{\mathbf{D}}_{k'} = \text{diag}((\tilde{\mathbf{Z}}_{\bullet k'})^T)$. Then

$$[\tilde{\mathbf{D}}_{k'}]_{(ij)(i'j')} = \begin{cases} \tau_{i'k'}(y_{i'j'}; \boldsymbol{\psi}) & \text{if } (ij) = (i'j') \\ 0 & \text{if } (ij) \neq (i'j'). \end{cases}$$

and

$$[\tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'}]_{(ij)(i'j')} = \begin{cases} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{i'k'}(y_{i'j'}; \boldsymbol{\psi}) & \text{if } (ij) = (i'j') \\ 0 & \text{if } (ij) \neq (i'j') \end{cases}$$

Therefore

$$[\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T - \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} + \delta_{kk'} \tilde{\mathbf{D}}_{k'}]_{(ij)(i'j')} = E_{\mathbf{Z}|\mathbf{y};\boldsymbol{\psi}}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]_{(ij)(i'j')}$$

This proves Proposition 6.3.1. □

Proposition 6.3.2.

$$E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}] = \tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(ij)\bullet} - \tau_{ik}(y_{ij};\psi)[\mathbf{I}_n]_{\bullet(ij)}\tilde{\mathbf{Z}}_{(ij)\bullet} + [\mathbf{I}_n]_{\bullet(ij)}[\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{k\bullet}$$

where \mathbf{I}_n is the $n \times n$ identity matrix, and $[\mathbf{I}_n]_{\bullet(ij)}$ is its $(ij)^{\text{th}}$ column (with the rows and columns of \mathbf{I}_n having the same labels as the rows of \mathbf{Z}).

Proof of Proposition 6.3.2. We show that the matrices given by $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}]$ and $\tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(ij)\bullet}$ differ only in their $(ij)^{\text{th}}$ row, and that the $(ij)^{\text{th}}$ row of $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}]$ is equal to $[\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{k\bullet}$ while the $(ij)^{\text{th}}$ row of $\tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(ij)\bullet}$ is equal to $\tau_{ik}(y_{ij};\psi)\tilde{\mathbf{Z}}_{(ij)\bullet}$.

The matrix $\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}$ is $n \times t$ and the element in row $(i'j')$ and column k' is denoted by

$$[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}]_{(i'j')k'} = z_{i'j'k} z_{ijk'}.$$

However, by definition of \mathbf{Z} ,

$$z_{i'j'k} z_{ijk'} = \begin{cases} 0 & \text{if } (ij) = (i'j') \text{ and } k \neq k' \\ z_{ijk} & \text{if } (ij) = (i'j') \text{ and } k = k' \\ z_{i'j'k} z_{ijk'} & \text{if } (ij) \neq (i'j') \text{ for all } k \text{ and } k' \end{cases}$$

Therefore,

$$E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}]_{(i'j')k'} = \begin{cases} 0, & \text{if } (ij) = (i'j') \text{ and } k \neq k' \\ \tau_{ik}(y_{ij};\psi), & \text{if } (ij) = (i'j') \text{ and } k = k' \\ \tau_{i'k}(y_{i'j'};\psi) \tau_{ik'}(y_{ij};\psi), & \text{if } (ij) \neq (i'j'), \text{ (by} \\ & \text{independence of individuals).} \end{cases}$$

Therefore, the $(ij)^{\text{th}}$ row of $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}]$ is equal to $[\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{k\bullet}$.

The matrix $\tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(ij)\bullet}$ is also $n \times t$ and the element in row $(i'j')$ and column k' is given by

$$[\tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(ij)\bullet}]_{(i'j')k'} = \tau_{i'k}(y_{i'j'};\psi) \tau_{ik'}(y_{ij};\psi),$$

which implies that the $(ij)^{\text{th}}$ row of $\tilde{\mathbf{Z}}_{\bullet k} \tilde{\mathbf{Z}}_{(ij)\bullet}$ is equal to $\tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tilde{\mathbf{Z}}_{(ij)\bullet}$. The result stated in Proposition 6.3.2 is obtained by using the $(ij)^{\text{th}}$ row of the $n \times n$ identity matrix as a device to modify the $(ij)^{\text{th}}$ row of $\tilde{\mathbf{Z}}_{\bullet k} \tilde{\mathbf{Z}}_{(ij)\bullet}$, thereby yielding the desired expectation. \square

Proposition 6.3.3.

$$E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[(\mathbf{Z}_{(ij)\bullet})^T \mathbf{Z}_{(i'j')\bullet}] = (1 - \delta_{(ij)(i'j')})(\tilde{\mathbf{Z}}_{(ij)\bullet})^T \tilde{\mathbf{Z}}_{(i'j')\bullet} + \delta_{(ij)(i'j')} \text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})$$

where $\delta_{(ij)(i'j')}$ is the Kronecker delta, which has value one if $(ij) = (i'j')$ and zero otherwise.

Proof of Proposition 6.3.3. The matrix $(\mathbf{Z}_{(ij)\bullet})^T \mathbf{Z}_{(i'j')\bullet}$ is $t \times t$ and the element in row k and column k' is given by

$$\begin{aligned} [(\mathbf{Z}_{(ij)\bullet})^T \mathbf{Z}_{(i'j')\bullet}]_{kk'} &= z_{ijk} z_{i'j'k'} \\ &= \begin{cases} 0 & \text{if } (ij) = (i'j') \text{ and } k \neq k' \\ z_{ijk} & \text{if } (ij) = (i'j') \text{ and } k = k' \\ z_{ijk} z_{i'j'k'} & \text{if } (ij) \neq (i'j') \text{ for all } k \text{ and } k' \end{cases} \end{aligned}$$

Therefore,

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[(\mathbf{Z}_{(ij)\bullet})^T \mathbf{Z}_{(i'j')\bullet}]_{kk'} &= \begin{cases} 0, & \text{if } (ij) = (i'j') \text{ and } k \neq k' \\ \tau_{ik}(y_{ij}; \boldsymbol{\psi}), & \text{if } (ij) = (i'j') \text{ and } k = k' \\ \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{i'k'}(y_{i'j'}; \boldsymbol{\psi}), & \text{if } (ij) \neq (i'j'), \text{ (by} \\ & \text{independence of individuals).} \end{cases} \\ &= \begin{cases} [\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{kk'} & \text{if } (ij) = (i'j') \\ [(\tilde{\mathbf{Z}}_{(ij)\bullet})^T \tilde{\mathbf{Z}}_{(i'j')\bullet}]_{kk'} & \text{if } (ij) \neq (i'j'). \end{cases} \end{aligned}$$

This proves Proposition 6.3.3. \square

6.4 Conditional Expectations of Outer Products of the Score Vectors

We need to calculate the missing-data (conditional) information matrix, $\mathcal{I}_m(\boldsymbol{\psi}; \mathbf{y})$ where

$$\mathcal{I}_m(\boldsymbol{\psi}; \mathbf{y}) = \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}}, \mathcal{U}_{\boldsymbol{\psi}}] = E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}} \mathcal{U}_{\boldsymbol{\psi}}^T] - (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}}])(E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}}])^T.$$

Formulae for calculating the components of $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}}]$ are already displayed in Equations (6.1) to (6.4). Table 6.2, below, lists the calculations that are dealt with in this section. These calculations are concerned with finding a formula for each block of the upper triangle of the symmetric matrix $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}} \mathcal{U}_{\boldsymbol{\psi}}^T]$. These formulae are key to making the calculation of $\mathcal{I}_m(\boldsymbol{\psi}; \mathbf{y})$ tractable and practical to implement.

Table 6.2: List of propositions that deal with the calculation of each block of the upper triangle of the symmetric matrix $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\boldsymbol{\psi}} \mathcal{U}_{\boldsymbol{\psi}}^T]$.

Component (block)	Proposition that reveals a formula for this component
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}}^T]$	Proposition 6.4.1
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)} \mathcal{U}_{(\sigma^2)}^T]$	Proposition 6.4.2
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\phi} \mathcal{U}_{\phi}^T]$	Proposition 6.4.3
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{(\sigma^2)}^T]$	Proposition 6.4.4
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\phi}^T]$	Proposition 6.4.5
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}, \mathcal{U}_{\phi}]$	Proposition 6.4.6
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}^*}^T]$	Proposition 6.4.7
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\mathbf{b}^*}^T]$	Proposition 6.4.8
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{(\sigma^2)}^T]$	Proposition 6.4.9
$E_{\mathbf{Z} \mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\phi}^T]$	Proposition 6.4.10

Proposition 6.4.1.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}}\mathcal{U}_{\mathbf{b}}^T] &= (E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}}]) (E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}}])^T + \frac{1}{\sigma^4} \mathbf{C}^T \left[\text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) \right. \\
&\quad + \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T) - 2 \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) \right) \\
&\quad \left. - \left(\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right) \left(\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right)^T \right] \mathbf{C}
\end{aligned}$$

Proof of Proposition 6.4.1.

$$\begin{aligned}
\mathcal{U}_{\mathbf{b}}\mathcal{U}_{\mathbf{b}}^T &= \frac{1}{\sigma^4} \mathbf{C}^T (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b} - \mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b} - \mathbf{Z}^T \ddot{\mathbf{y}})^T \mathbf{C} \\
&= \frac{1}{\sigma^4} \mathbf{C}^T \left[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T - (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z}^T \ddot{\mathbf{y}})^T \right. \\
&\quad \left. - (\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T + (\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z}^T \ddot{\mathbf{y}})^T \right] \mathbf{C} \quad (6.49)
\end{aligned}$$

Therefore, we need the expectations of the three $t \times t$ matrices

1. $(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T$
2. $(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z}^T \ddot{\mathbf{y}})^T$
3. $(\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z}^T \ddot{\mathbf{y}})^T$

First we find the $(kk')^{\text{th}}$ element of each matrix.

Now $\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}$ and $\mathbf{Z}^T \ddot{\mathbf{y}}$ are column vectors of order t . For simplicity let $\boldsymbol{\mu} = \mathbf{C} \mathbf{b}$ and $\mu_k = \mathbf{C}_{k\bullet} \mathbf{b}$.

Therefore, $\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b} = \text{diag}(\mathbf{1}_n^T \mathbf{Z}) \mathbf{C} \mathbf{b} = \text{diag}(\mathbf{1}_n^T \mathbf{Z}) \boldsymbol{\mu}$.

The k^{th} element of the column vectors $\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}$ and $\mathbf{Z}^T \ddot{\mathbf{y}}$ are, respectively,

$$\begin{aligned}
[\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}]_{k\bullet} &= \mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mathbf{C}_{k\bullet} \mathbf{b} = \mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k \\
[\mathbf{Z}^T \ddot{\mathbf{y}}]_{k\bullet} &= [\mathbf{Z}^T]_{k\bullet} \ddot{\mathbf{y}} = (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}}
\end{aligned}$$

and, clearly, the k^{th} element is a scalar in each case.

$$\begin{aligned}
1. \quad & [(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T]_{kk'} = [\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}]_{k\bullet} [(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T]_{\bullet k'} \\
& = (\mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k) (\mu_{k'} (\mathbf{Z}_{\bullet k'})^T \mathbf{1}_n) \\
& = \mathbf{1}_n^T \mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T \mathbf{1}_n \mu_k \mu_{k'}
\end{aligned}$$

since μ_k and $\mu_{k'}$ are scalars.

$$\begin{aligned}
2. \quad & [(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{kk'} = [\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}]_{k\bullet} [(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{\bullet k'} \\
& = (\mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k) (\ddot{\mathbf{y}}^T \mathbf{Z}_{\bullet k'}) \\
& = (\mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k) (\mathbf{Z}_{\bullet k'})^T \ddot{\mathbf{y}}
\end{aligned}$$

since $\ddot{\mathbf{y}}^T \mathbf{Z}_{\bullet k'}$ is a scalar and so is symmetric.

$$= \mathbf{1}_n^T \mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T \ddot{\mathbf{y}} \mu_k, \text{ since } \mu_k \text{ is a scalar.}$$

$$\begin{aligned}
3. \quad & [(\mathbf{Z}^T \ddot{\mathbf{y}})(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{kk'} = [\mathbf{Z}^T \ddot{\mathbf{y}}]_{k\bullet} [(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{\bullet k'} \\
& = (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}} (\ddot{\mathbf{y}}^T \mathbf{Z}_{\bullet k'}) \\
& = \ddot{\mathbf{y}}^T \mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T \ddot{\mathbf{y}}, \text{ since } (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}} \text{ and } \ddot{\mathbf{y}}^T \mathbf{Z}_{\bullet k'} \text{ are}
\end{aligned}$$

both scalars and so are both symmetric.

Hence, we have the following simplifications:

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T]_{kk'} &= \mathbf{1}_n^T E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T] \mathbf{1}_n \mu_k \mu_{k'} \\
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{kk'} &= \mathbf{1}_n^T E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T] \ddot{\mathbf{y}} \mu_k \\
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T \ddot{\mathbf{y}})(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{kk'} &= \ddot{\mathbf{y}}^T E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T] \ddot{\mathbf{y}}
\end{aligned}$$

A formula for evaluating $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T]$ is given in Proposition 6.3.1. Therefore we now have the $(kk')^{\text{th}}$ element of the expectations of the required matrices. Next, we inspect that element in each case and show that in every case it can be written as the $(kk')^{\text{th}}$ element of another matrix. Then we invoke the rule of matrix equality, so revealing a matrix formula for evaluating the required expectations.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T]_{kk'} &= \mathbf{1}_n^T E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T] \mathbf{1}_n \mu_k \mu_{k'} \\
&= \mathbf{1}_n^T [\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T - \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} + \delta_{kk'} \tilde{\mathbf{D}}_{k'}] \mathbf{1}_n \mu_k \mu_{k'} \\
&= \mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T \mathbf{1}_n \mu_k \mu_{k'} - \mathbf{1}_n^T \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} \mathbf{1}_n \mu_k \mu_{k'} + \delta_{kk'} \mathbf{1}_n^T \tilde{\mathbf{D}}_{k'} \mathbf{1}_n \mu_k \mu_{k'} \\
&= [\mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k} \mu_k][(\tilde{\mathbf{Z}}_{\bullet k'})^T \mathbf{1}_n \mu_{k'}] - [\mathbf{1}_n^T \tilde{\mathbf{D}}_k \mu_k][\mu_{k'} \tilde{\mathbf{D}}_{k'} \mathbf{1}_n] + \delta_{kk'} [\mathbf{1}_n^T \tilde{\mathbf{D}}_k \mathbf{1}_n \mu_{k'}^2]
\end{aligned}$$

since μ_k and $\mu_{k'}$ are scalars and $\tilde{\mathbf{D}}_k = \tilde{\mathbf{D}}_{k'}$, $\mu_k = \mu_{k'}$ when $k = k'$.

Using the identities given in Equations (6.20) to (6.24), we obtain the following simplification.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T]_{kk'} &= [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu}]_{k\bullet} [\boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})]_{\bullet k'} - [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T]_{k\bullet} [\tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T)]_{\bullet k'} \\
&\quad + \delta_{kk'} [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})]_{k\bullet} [\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T)]_{\bullet k'} \\
&= [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})]_{kk'} - [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T)]_{kk'} \\
&\quad + [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T)]_{kk'}
\end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T] &= \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \\
&\quad - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \\
&\quad + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T). \tag{6.50}
\end{aligned}$$

We proceed in a similar way to evaluate the next expectation.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{kk'} &= \mathbf{1}_n^T E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T] \ddot{\mathbf{y}} \mu_k \\
&= \mathbf{1}_n^T [\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T - \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} + \delta_{kk'} \tilde{\mathbf{D}}_{k'}] \ddot{\mathbf{y}} \mu_k \\
&= \mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T \ddot{\mathbf{y}} \mu_k - \mathbf{1}_n^T \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}} \mu_k + \delta_{kk'} \mathbf{1}_n^T \tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}} \mu_k \\
&= [\mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k} \mu_k][(\tilde{\mathbf{Z}}_{\bullet k'})^T \ddot{\mathbf{y}}] - [\mathbf{1}_n^T \tilde{\mathbf{D}}_k \mu_k][\tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}}] + \delta_{kk'} [\mathbf{1}_n^T \tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}} \mu_k]
\end{aligned}$$

Using the identities given in Equations (6.20), (6.22), (6.25), (6.26) and (6.27), we obtain the following simplification.

$$\begin{aligned}
& E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{kk'} \\
&= [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu}]_{k\bullet} [\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}]_{\bullet k'} - [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T]_{k\bullet} [\text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{\bullet k'} \\
&\quad + \delta_{kk'} [\text{diag}(\boldsymbol{\mu}^T)]_{k\bullet} [\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}})]_{\bullet k'} \\
&= [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}]_{kk'} - [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{kk'} + \delta_{kk'} [\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}})]_{kk'}
\end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})(\mathbf{Z}^T \ddot{\mathbf{y}})^T] &= \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \\
&\quad + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}})
\end{aligned} \tag{6.51}$$

and taking the transpose of this we obtain

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \ddot{\mathbf{y}})(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T] &= \tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \\
&\quad + \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T).
\end{aligned} \tag{6.52}$$

$$\begin{aligned}
& E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \ddot{\mathbf{y}})(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{kk'} \\
&= \ddot{\mathbf{y}}^T E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T] \ddot{\mathbf{y}} \\
&= \ddot{\mathbf{y}}^T [\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T - \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} + \delta_{kk'} \tilde{\mathbf{D}}_{k'}] \ddot{\mathbf{y}} \\
&= \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T \ddot{\mathbf{y}} - \ddot{\mathbf{y}}^T \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}} + \delta_{kk'} \ddot{\mathbf{y}}^T \tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}} \\
&= [\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}_{\bullet k}][(\tilde{\mathbf{Z}}_{\bullet k'})^T \ddot{\mathbf{y}}] - [\ddot{\mathbf{y}}^T \tilde{\mathbf{D}}_k][\tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}}] + \delta_{kk'} [\ddot{\mathbf{y}}^T \tilde{\mathbf{D}}_{k'} \ddot{\mathbf{y}}]
\end{aligned}$$

Using the identities given in Equations (6.25), (6.26), (6.28), (6.29) and (6.30), we obtain the following simplification.

$$\begin{aligned}
& E_{\mathbf{Z}|\mathbf{y}; \psi}[(\mathbf{Z}^T \ddot{\mathbf{y}})(\mathbf{Z}^T \ddot{\mathbf{y}})^T]_{kk'} \\
&= [\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}}]_{k\bullet} [\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}]_{\bullet k'} - [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T)]_{k\bullet} [\text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{\bullet k'} + \delta_{kk'} [\text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}})]_{k'k'} \\
&= [\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}]_{kk'} - [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{kk'} + [\text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}})]_{kk'}
\end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T \ddot{\mathbf{y}})(\mathbf{Z}^T \ddot{\mathbf{y}})^T] &= \tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \\ &\quad + \text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}). \end{aligned} \quad (6.53)$$

After making the relevant substitutions we obtain the following result.

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}}^T] &= \frac{1}{\sigma^4} \mathbf{C}^T \left[\left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \right. \right. \\ &\quad - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T) \Big) \\ &\quad - \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) \right) \\ &\quad - \left(\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) + \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T) \right) \\ &\quad \left. + \left(\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} + \text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) \right) \right] \mathbf{C} \end{aligned}$$

Simplifying the above expression yields the result of Proposition 6.4.1. \square

Proposition 6.4.2.

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)} \mathcal{U}_{(\sigma^2)}^T] &= (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}])^T \\ &\quad + \frac{1}{4\sigma^8} \boldsymbol{\mu}^T \left[4 \text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) - 4 \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \right. \\ &\quad \left. + 4 \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \right) \right. \\ &\quad \left. + \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \right) \text{diag}(\boldsymbol{\mu}^T) \right] \boldsymbol{\mu} \end{aligned}$$

Proof of Proposition 6.4.2.

$$\begin{aligned} \mathcal{U}_{(\sigma^2)} \mathcal{U}_{(\sigma^2)}^T &= \left[-\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} (\ddot{\mathbf{y}}^T \ddot{\mathbf{y}} - 2\boldsymbol{\mu}^T \mathbf{Z}^T \ddot{\mathbf{y}} + \boldsymbol{\mu}^T \mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu}) \right]^2 \\ &= \frac{n^2}{4\sigma^4} - \frac{2n}{4\sigma^6} (\ddot{\mathbf{y}}^T \ddot{\mathbf{y}} - 2\boldsymbol{\mu}^T \mathbf{Z}^T \ddot{\mathbf{y}} + \boldsymbol{\mu}^T \mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu}) \\ &\quad + \frac{1}{4\sigma^8} \left[(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}})^2 + 4\boldsymbol{\mu}^T (\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z}^T \ddot{\mathbf{y}})^T \boldsymbol{\mu} + \boldsymbol{\mu}^T (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T \boldsymbol{\mu} \right. \\ &\quad \left. - 4\boldsymbol{\mu}^T (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z}^T \ddot{\mathbf{y}})^T \boldsymbol{\mu} - 4(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \ddot{\mathbf{y}}^T \mathbf{Z} \boldsymbol{\mu} + 2(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \boldsymbol{\mu}^T \mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu} \right] \end{aligned} \quad (6.54)$$

The expectations needed to calculate $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}\mathcal{U}_{(\sigma^2)}^T]$ are known from Equations (5.48) to (5.55) and from Equations (6.50) to (6.53). It only remains to make the substitutions and simplify the resulting expression. On making the substitutions, we obtain:

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}\mathcal{U}_{(\sigma^2)}^T] &= \frac{n^2}{4\sigma^4} - \frac{2n}{4\sigma^6} \left(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}} - 2\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} + \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \right) + \frac{1}{4\sigma^8} \left[(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}})^2 \right. \\
&+ 4\boldsymbol{\mu}^T \left(\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} + \text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) \right) \boldsymbol{\mu} \\
&+ \boldsymbol{\mu}^T \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \right. \\
&\quad \left. \left. + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T) \right) \boldsymbol{\mu} \right. \\
&- 4\boldsymbol{\mu}^T \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) \right) \boldsymbol{\mu} \\
&\left. - 4(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} \boldsymbol{\mu} + 2(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \right]
\end{aligned}$$

After simplifying the above expression, we obtain the result of Proposition 6.4.2. \square

Proposition 6.4.3.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi \mathcal{U}_\phi^T] &= (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi])^T \\
&+ \sum_{i=1}^s \sum_{i'=1}^s \left[\left(\frac{\partial}{\partial \phi} \mathbf{h}_i^T(\phi) \right) \left(\text{diag}(\mathbf{1}_{n_{i'}}^T \boldsymbol{\Delta}_{i'} \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i) \right. \right. \\
&\quad \left. \left. - \tilde{\mathbf{Z}}_i^T \boldsymbol{\Delta}_i \boldsymbol{\Delta}_{i'}^T \tilde{\mathbf{Z}}_{i'} \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_{i'}(\phi) \right) \right].
\end{aligned}$$

Proof of Proposition 6.4.3.

$$\begin{aligned}
\mathcal{U}_\phi \mathcal{U}_\phi^T &= \left[\sum_{i=1}^s \left(\frac{\partial}{\partial \phi} \mathbf{h}_i^T(\phi) \right) \mathbf{Z}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i} \right] \left[\sum_{i'=1}^s \mathbf{1}_{n_{i'}}^T \boldsymbol{\Delta}_{i'} \mathbf{Z} \left(\frac{\partial}{\partial \phi} \mathbf{h}_{i'}(\phi) \right) \right] \\
&= \sum_{i=1}^s \sum_{i'=1}^s \left(\frac{\partial}{\partial \phi} \mathbf{h}_i^T(\phi) \right) \mathbf{Z}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \boldsymbol{\Delta}_{i'} \mathbf{Z} \left(\frac{\partial}{\partial \phi} \mathbf{h}_{i'}(\phi) \right) \quad (6.55)
\end{aligned}$$

First, we find the $(kk')^{\text{th}}$ element of the $t \times t$ matrix: $\mathbf{Z}^T \boldsymbol{\Delta}_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \boldsymbol{\Delta}_{i'} \mathbf{Z}$.

The k^{th} element of the column vector $\mathbf{Z}^T \Delta_i^T \mathbf{1}_{n_i}$ is equal to

$$[\mathbf{Z}^T \Delta_i^T \mathbf{1}_{n_i}]_{k\bullet} = (\mathbf{Z}^T)_{k\bullet} \Delta_i^T \mathbf{1}_{n_i} = (\mathbf{Z}_{\bullet k})^T \Delta_i^T \mathbf{1}_{n_i}$$

The k'^{th} element of the row vector $\mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}$ is equal to

$$[\mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}]_{\bullet k'} = \mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}_{\bullet k'}$$

Therefore,

$$\begin{aligned} [\mathbf{Z}^T \Delta_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}]_{kk'} &= [\mathbf{Z}^T \Delta_i^T \mathbf{1}_{n_i}]_{k\bullet} [\mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}]_{\bullet k'} \\ &= (\mathbf{Z}_{\bullet k})^T \Delta_i^T \mathbf{1}_{n_i} (\mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}_{\bullet k'}) \\ &= \mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T \Delta_{i'}^T \mathbf{1}_{n_{i'}} \end{aligned}$$

since $(\mathbf{Z}_{\bullet k})^T \Delta_i^T \mathbf{1}_{n_i}$ and $\mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}_{\bullet k'}$ are scalars.

A formula for evaluating $E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]$ is given in Proposition 6.3.1. Therefore we have the following simplifications.

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z}^T \Delta_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}]_{kk'} &= \mathbf{1}_{n_i}^T \Delta_i [\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T - \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} + \delta_{kk'} \tilde{\mathbf{D}}_{k'}] \Delta_{i'}^T \mathbf{1}_{n_{i'}} \\ &= \mathbf{1}_{n_i}^T \Delta_i \tilde{\mathbf{Z}}_{\bullet k} (\tilde{\mathbf{Z}}_{\bullet k'})^T \Delta_{i'}^T \mathbf{1}_{n_{i'}} - \mathbf{1}_{n_i}^T \Delta_i \tilde{\mathbf{D}}_k \tilde{\mathbf{D}}_{k'} \Delta_{i'}^T \mathbf{1}_{n_{i'}} + \delta_{kk'} \mathbf{1}_{n_i}^T \Delta_i \tilde{\mathbf{D}}_{k'} \Delta_{i'}^T \mathbf{1}_{n_{i'}} \end{aligned}$$

Using the identities given in Equations (6.31) to (6.35), we obtain a further simplification.

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z}^T \Delta_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}]_{kk'} &= [\tilde{\mathbf{Z}}_i^T \mathbf{1}_{n_i}]_{k\bullet} [\mathbf{1}_{n_{i'}}^T \tilde{\mathbf{Z}}_{i'}]_{\bullet k'} - [\tilde{\mathbf{Z}}_i^T \Delta_i]_{k\bullet} [\Delta_{i'}^T \tilde{\mathbf{Z}}_{i'}]_{\bullet k'} + \delta_{kk'} [\text{diag}(\mathbf{1}_{n_{i'}}^T \Delta_{i'} \Delta_i^T \tilde{\mathbf{Z}}_i)]_{k'k'} \\ &= [\tilde{\mathbf{Z}}_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \tilde{\mathbf{Z}}_{i'}]_{kk'} - [\tilde{\mathbf{Z}}_i^T \Delta_i \Delta_{i'}^T \tilde{\mathbf{Z}}_{i'}]_{kk'} + [\text{diag}(\mathbf{1}_{n_{i'}}^T \Delta_{i'} \Delta_i^T \tilde{\mathbf{Z}}_i)]_{kk'} \end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z}^T \Delta_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \Delta_{i'} \mathbf{Z}] &= \tilde{\mathbf{Z}}_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \tilde{\mathbf{Z}}_{i'} - \tilde{\mathbf{Z}}_i^T \Delta_i \Delta_{i'}^T \tilde{\mathbf{Z}}_{i'} + \text{diag}(\mathbf{1}_{n_{i'}}^T \Delta_{i'} \Delta_i^T \tilde{\mathbf{Z}}_i). \end{aligned} \quad (6.56)$$

$$E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi \mathcal{U}_\phi^T] = \sum_{i=1}^s \sum_{i'=1}^s \left[\left(\frac{\partial}{\partial \phi} \mathbf{h}_i^T(\phi) \right) \left(\tilde{\mathbf{Z}}_i^T \mathbf{1}_{n_i} \mathbf{1}_{n_{i'}}^T \tilde{\mathbf{Z}}_{i'} - \tilde{\mathbf{Z}}_i^T \Delta_i \Delta_{i'}^T \tilde{\mathbf{Z}}_{i'} \right. \right. \\ \left. \left. + \text{diag}(\mathbf{1}_{n_{i'}}^T \Delta_{i'} \Delta_i^T \tilde{\mathbf{Z}}_i) \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_{i'}^T(\phi) \right) \right]$$

Expanding and re-grouping the terms in the above expression gives the result of Proposition 6.4.3. \square

Proposition 6.4.4.

$$E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{(\sigma^2)}^T] = (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}])^T \\ + \frac{1}{2\sigma^6} \mathbf{C}^T \left[2 \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \right) \right. \\ - 2 \text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) + 2 \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \\ - \text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \right) \text{diag}(\boldsymbol{\mu}^T) \\ \left. + \left(\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \right) \text{diag}(\boldsymbol{\mu}^T) \right] \boldsymbol{\mu}$$

Proof of Proposition 6.4.4.

$$\mathcal{U}_{\mathbf{b}} \mathcal{U}_{(\sigma^2)}^T = -\frac{1}{\sigma^2} \mathbf{C}^T \left[-\frac{n}{2\sigma^2} \left(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b} - \mathbf{Z}^T \ddot{\mathbf{y}} \right) \right. \\ + \frac{1}{2\sigma^4} \left[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) - 2(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z}^T \ddot{\mathbf{y}})^T \boldsymbol{\mu} \right. \\ + (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T \boldsymbol{\mu} - (\mathbf{Z}^T \ddot{\mathbf{y}}) (\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \\ \left. \left. + 2(\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z}^T \ddot{\mathbf{y}})^T \boldsymbol{\mu} - (\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})^T \boldsymbol{\mu} \right] \right] \quad (6.57)$$

The expectations needed to calculate $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{(\sigma^2)}^T]$ are known from Equations (5.48) to (5.55) and from Equations (6.50) to (6.53). It only remains to make the substitutions and simplify the resulting expression.

On making the substitutions, we obtain:

$$\begin{aligned}
& E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{(\sigma^2)}^T] \\
&= -\frac{1}{\sigma^2} \mathbf{C}^T \left[-\frac{n}{2\sigma^2} \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} - \tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \right) + \frac{1}{2\sigma^4} \left[\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} (\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \right. \right. \\
&\quad - 2 \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) \right) \boldsymbol{\mu} \\
&\quad + \left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \right. \\
&\quad \quad \left. \left. + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T) \right) \boldsymbol{\mu} - (\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}}) (\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \right. \\
&\quad \left. + 2 \left(\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}} - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} + \text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}) \right) \boldsymbol{\mu} \right. \\
&\quad \left. \left. - \left(\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \boldsymbol{\mu}^T \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) + \text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}) \text{diag}(\boldsymbol{\mu}^T) \right) \boldsymbol{\mu} \right] \right]
\end{aligned}$$

After simplifying the above expression, we obtain the result of Proposition 6.4.4. \square

Proposition 6.4.5.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\phi}^T] &= (E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}}]) (E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\phi}])^T \\
&\quad - \frac{1}{\sigma^2} \mathbf{C}^T \sum_{i=1}^s \left[\left(\text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i \right. \right. \\
&\quad \left. \left. - \text{diag}(\ddot{\mathbf{y}}^T \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i) + \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right]
\end{aligned}$$

Proof of Proposition 6.4.5.

$$\begin{aligned}
\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\phi}^T &= -\frac{1}{\sigma^2} \mathbf{C}^T (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b} - \mathbf{Z}^T \ddot{\mathbf{y}}) \left[\sum_{i=1}^s \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z} \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right] \\
&= -\frac{1}{\sigma^2} \mathbf{C}^T \sum_{i=1}^s \left((\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z} - (\mathbf{Z}^T \ddot{\mathbf{y}}) \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z} \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \quad (6.58)
\end{aligned}$$

Therefore, we need the expectations of the two $t \times t$ matrices

1. $(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z}$
2. $(\mathbf{Z}^T \ddot{\mathbf{y}}) \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z}$

The k^{th} element of the column vectors $(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b})$ and $(\mathbf{Z}^T \ddot{\mathbf{y}})$ and the k'^{th} element of the row vector $\mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}$ are given below.

$$\begin{aligned} [\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}]_{k\bullet} &= \mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mathbf{C}_{k\bullet} \mathbf{b} = \mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k \\ [\mathbf{Z}^T \ddot{\mathbf{y}}]_{k\bullet} &= [\mathbf{Z}^T]_{k\bullet} \ddot{\mathbf{y}} = (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}} \\ [\mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}]_{\bullet k'} &= \mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}_{\bullet k'} \end{aligned}$$

The next step is to inspect the kk'^{th} element of the required matrices.

$$\begin{aligned} 1. [(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) \mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}]_{kk'} &= [\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}]_{k\bullet} [\mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}]_{\bullet k'} \\ &= (\mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k) (\mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}_{\bullet k'}) \\ &= \mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k (\mathbf{Z}_{\bullet k'})^T \Delta_i^T \mathbf{1}_{n_i} \\ &\quad \text{since } \mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}_{\bullet k'} \text{ is a scalar} \\ &= \mathbf{1}_n^T \mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T \Delta_i^T \mathbf{1}_{n_i} \mu_k, \text{ since } \mu_k \text{ is a scalar.} \\ 2. [(\mathbf{Z}^T \ddot{\mathbf{y}}) \mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}]_{kk'} &= [\mathbf{Z}^T \ddot{\mathbf{y}}]_{k\bullet} [\mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}]_{\bullet k'} \\ &= (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}} (\mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}_{\bullet k'}) \\ &= \ddot{\mathbf{y}}^T \mathbf{Z}_{\bullet k} (\mathbf{Z}_{\bullet k'})^T \Delta_i^T \mathbf{1}_{n_i}, \text{ since } (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}} \text{ and } \mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}_{\bullet k'} \\ &\quad \text{are both scalars and so are both symmetric.} \end{aligned}$$

Now, we take the expectations of the above expressions and substitute the formula for $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]$ which is given in Proposition 6.3.1, obtaining the expectation of the kk'^{th} element in each case.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\mathbf{ZC}\mathbf{b})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}]_{kk'} \\
&= \mathbf{1}_n^T E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]\Delta_i^T\mathbf{1}_{n_i}\mu_k \\
&= \mathbf{1}_n^T[\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T - \tilde{\mathbf{D}}_k\tilde{\mathbf{D}}_{k'} + \delta_{kk'}\tilde{\mathbf{D}}_{k'}]\Delta_i^T\mathbf{1}_{n_i}\mu_k \\
&= \mathbf{1}_n^T\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T\Delta_i^T\mathbf{1}_{n_i}\mu_k - \mathbf{1}_n^T\tilde{\mathbf{D}}_k\tilde{\mathbf{D}}_{k'}\Delta_i^T\mathbf{1}_{n_i}\mu_k + \delta_{kk'}\mathbf{1}_n^T\tilde{\mathbf{D}}_{k'}\Delta_i^T\mathbf{1}_{n_i}\mu_k \\
&= [\mathbf{1}_n^T\tilde{\mathbf{Z}}_{\bullet k}\mu_k][(\tilde{\mathbf{Z}}_{\bullet k'})^T\Delta_i^T\mathbf{1}_{n_i}] - [\mathbf{1}_n^T\tilde{\mathbf{D}}_k\mu_k][\tilde{\mathbf{D}}_{k'}\Delta_i^T\mathbf{1}_{n_i}] + \delta_{kk'}[\mathbf{1}_n^T\tilde{\mathbf{D}}_{k'}\mu_k\Delta_i^T\mathbf{1}_{n_i}] \\
&\quad \text{since } \mu_k \text{ is a scalar and } \tilde{\mathbf{D}}_k = \tilde{\mathbf{D}}_{k'} \text{ when } k = k'.
\end{aligned}$$

Using the identities given in Equations (6.20), (6.22), (6.32), (6.34) and (6.36) we obtain the following simplification.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\mathbf{ZC}\mathbf{b})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}]_{kk'} \\
&= [\text{diag}(\mathbf{1}_n^T\tilde{\mathbf{Z}})\boldsymbol{\mu}]_{k\bullet}[\mathbf{1}_{n_i}^T\tilde{\mathbf{Z}}_i]_{\bullet k'} - [\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T]_{k\bullet}[\Delta_i^T\tilde{\mathbf{Z}}_i]_{\bullet k'} \\
&\quad + \delta_{kk'}\left[\text{diag}(\mathbf{1}_{n_i}^T\tilde{\mathbf{Z}}_i\text{diag}(\boldsymbol{\mu}^T))\right]_{kk} \\
&= [\text{diag}(\mathbf{1}_n^T\tilde{\mathbf{Z}})\boldsymbol{\mu}\mathbf{1}_{n_i}^T\tilde{\mathbf{Z}}_i]_{kk'} - [\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T\Delta_i^T\tilde{\mathbf{Z}}_i]_{kk'} + \left[\text{diag}(\mathbf{1}_{n_i}^T\tilde{\mathbf{Z}}_i\text{diag}(\boldsymbol{\mu}^T))\right]_{kk'}
\end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\mathbf{ZC}\mathbf{b})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}] \\
&= \text{diag}(\mathbf{1}_n^T\tilde{\mathbf{Z}})\boldsymbol{\mu}\mathbf{1}_{n_i}^T\tilde{\mathbf{Z}}_i - \text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T\Delta_i^T\tilde{\mathbf{Z}}_i + \text{diag}(\mathbf{1}_{n_i}^T\tilde{\mathbf{Z}}_i\text{diag}(\boldsymbol{\mu}^T)) \\
&= \text{diag}(\mathbf{1}_n^T\tilde{\mathbf{Z}})\boldsymbol{\mu}\mathbf{1}_{n_i}^T\tilde{\mathbf{Z}}_i - \text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}_i^T\tilde{\mathbf{Z}}_i + \text{diag}(\mathbf{1}_{n_i}^T\tilde{\mathbf{Z}}_i\text{diag}(\boldsymbol{\mu}^T)) . \tag{6.59}
\end{aligned}$$

We evaluate the next expectation using the procedure employed above.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\ddot{\mathbf{y}})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}]_{kk'} \\
&= \ddot{\mathbf{y}}^T E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}(\mathbf{Z}_{\bullet k'})^T]\Delta_i^T\mathbf{1}_{n_i} \\
&= \ddot{\mathbf{y}}^T[\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T - \tilde{\mathbf{D}}_k\tilde{\mathbf{D}}_{k'} + \delta_{kk'}\tilde{\mathbf{D}}_{k'}]\Delta_i^T\mathbf{1}_{n_i} \\
&= \ddot{\mathbf{y}}^T\tilde{\mathbf{Z}}_{\bullet k}(\tilde{\mathbf{Z}}_{\bullet k'})^T\Delta_i^T\mathbf{1}_{n_i} - \ddot{\mathbf{y}}^T\tilde{\mathbf{D}}_k\tilde{\mathbf{D}}_{k'}\Delta_i^T\mathbf{1}_{n_i} + \delta_{kk'}\ddot{\mathbf{y}}^T\tilde{\mathbf{D}}_{k'}\Delta_i^T\mathbf{1}_{n_i} \\
&= [\ddot{\mathbf{y}}^T\tilde{\mathbf{Z}}_{\bullet k}][(\tilde{\mathbf{Z}}_{\bullet k'})^T\Delta_i^T\mathbf{1}_{n_i}] - [\ddot{\mathbf{y}}^T\tilde{\mathbf{D}}_k][\tilde{\mathbf{D}}_{k'}\Delta_i^T\mathbf{1}_{n_i}] + \delta_{kk'}[\ddot{\mathbf{y}}^T\tilde{\mathbf{D}}_{k'}\Delta_i^T\mathbf{1}_{n_i}]
\end{aligned}$$

Using the identities given in Equations (6.28), (6.29), (6.32), (6.34) and (6.37) we obtain the following simplification.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T \ddot{\mathbf{y}}) \mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}]_{kk'} &= [\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}}]_{k\bullet} [\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i]_{\bullet k'} - [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T)]_{k\bullet} [\Delta_i^T \tilde{\mathbf{Z}}_i]_{\bullet k'} + \delta_{kk'} [\text{diag}(\ddot{\mathbf{y}}^T \Delta_i^T \tilde{\mathbf{Z}}_i)]_{k'k'} \\
&= [\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i]_{kk'} - [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \Delta_i^T \tilde{\mathbf{Z}}_i]_{kk'} + [\text{diag}(\ddot{\mathbf{y}}^T \Delta_i^T \tilde{\mathbf{Z}}_i)]_{kk'}
\end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T \ddot{\mathbf{y}}) \mathbf{1}_{n_i}^T \Delta_i \mathbf{Z}] = \tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \Delta_i^T \tilde{\mathbf{Z}}_i + \text{diag}(\ddot{\mathbf{y}}^T \Delta_i^T \tilde{\mathbf{Z}}_i). \quad (6.60)$$

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\phi}^T] &= -\frac{1}{\sigma^2} \mathbf{C}^T \sum_{i=1}^s \left[\left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i \right. \right. \\
&\quad \left. \left. + \text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) \right) \right. \\
&\quad \left. - \left(\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \Delta_i^T \tilde{\mathbf{Z}}_i \right. \right. \\
&\quad \left. \left. + \text{diag}(\ddot{\mathbf{y}}^T \Delta_i^T \tilde{\mathbf{Z}}_i) \right) \right] \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \quad (6.61)
\end{aligned}$$

After simplifying the above expression, we obtain the result of Proposition 6.4.5. \square

Proposition 6.4.6.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}, \mathcal{U}_{\phi}] &= (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\phi}])^T \\
&\quad + \frac{1}{2\sigma^4} \boldsymbol{\mu}^T \sum_{i=1}^s \left[\left(\text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i \right. \right. \\
&\quad \left. \left. - 2 \text{diag}(\ddot{\mathbf{y}}^T \Delta_i^T \tilde{\mathbf{Z}}_i) + 2 \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \Delta_i^T \tilde{\mathbf{Z}}_i \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right]
\end{aligned}$$

Proof of Proposition 6.4.6.

$$\begin{aligned}
\mathcal{U}_{(\sigma^2)} \mathcal{U}_\phi^T &= \left[-\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4} (\ddot{\mathbf{y}}^T \ddot{\mathbf{y}} - 2\boldsymbol{\mu}^T \mathbf{Z}^T \ddot{\mathbf{y}} + \boldsymbol{\mu}^T \mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu}) \right] \left[\sum_{i=1}^s \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z} \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right] \\
&= -\frac{n}{2\sigma^2} \sum_{i=1}^s \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z} \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \\
&\quad + \frac{1}{2\sigma^4} \sum_{i=1}^s \left[\left((\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z} - 2\boldsymbol{\mu}^T (\mathbf{Z}^T \ddot{\mathbf{y}}) \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z} \right. \right. \\
&\quad \left. \left. + \boldsymbol{\mu}^T (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) \mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i \mathbf{Z} \right) \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right] \quad (6.62)
\end{aligned}$$

The expectations needed to calculate $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)} \mathcal{U}_\phi^T]$ are known from Equations (5.48) to (5.50) and from Equations (6.59) to (6.60). It only remains to make the substitutions and simplify the resulting expression.

On making the substitutions, we obtain:

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)} \mathcal{U}_\phi^T] &= -\frac{n}{2\sigma^2} \sum_{i=1}^s \mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) + \frac{1}{2\sigma^4} \sum_{i=1}^s \left\{ \left[(\ddot{\mathbf{y}}^T \ddot{\mathbf{y}}) \mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \right. \right. \\
&\quad - 2\boldsymbol{\mu}^T \left(\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i + \text{diag}(\ddot{\mathbf{y}}^T \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i) \right) \\
&\quad \left. + \boldsymbol{\mu}^T \left(\text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i + \text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) \right) \right] \\
&\quad \left. \times \left(\frac{\partial}{\partial \phi} \mathbf{h}_i(\phi) \right) \right\}
\end{aligned}$$

After simplifying the above expression, we obtain the result of Proposition 6.4.6. \square

Proposition 6.4.7.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}^*}^T] &= (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}])^T \\
&\quad + \frac{1}{\sigma^4} \mathbf{C}^T \left[\text{diag}(\boldsymbol{\mu}^T) \left(\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right) \right. \\
&\quad \left. - \left(\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \right] \mathbf{X}_2
\end{aligned}$$

Proof of Proposition 6.4.7.

$$\begin{aligned}
\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}^*}^T &= \frac{1}{\sigma^4} \mathbf{C}^T (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b} - \mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z} \mathbf{C} \mathbf{b} - \ddot{\mathbf{y}})^T \mathbf{X}_2 \\
&= \frac{1}{\sigma^4} \mathbf{C}^T \left[(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z} \mathbf{C} \mathbf{b})^T - (\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\ddot{\mathbf{y}})^T \right. \\
&\quad \left. - (\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z} \mathbf{C} \mathbf{b})^T + (\mathbf{Z}^T \ddot{\mathbf{y}}) (\ddot{\mathbf{y}})^T \right] \mathbf{X}_2
\end{aligned} \tag{6.63}$$

Now

$$E_{\mathbf{Z}|\mathbf{y}; \psi} [(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\ddot{\mathbf{y}})^T] = \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \ddot{\mathbf{y}}^T \tag{6.64}$$

and

$$E_{\mathbf{Z}|\mathbf{y}; \psi} [(\mathbf{Z}^T \ddot{\mathbf{y}}) (\ddot{\mathbf{y}})^T] = (\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}}) (\ddot{\mathbf{y}})^T. \tag{6.65}$$

We need the expectations of the two $t \times n$ matrices

1. $(\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}) (\mathbf{Z} \mathbf{C} \mathbf{b})^T = (\mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu}) (\mathbf{Z} \boldsymbol{\mu})^T$
2. $(\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z} \mathbf{C} \mathbf{b})^T = (\mathbf{Z}^T \ddot{\mathbf{y}}) (\mathbf{Z} \boldsymbol{\mu})^T$

First we find the element in row k and column (ij) of each matrix.

From previous calculations (see the proof of Proposition 6.4.1) we have that k^{th} element of the column vectors $\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}$ and $\mathbf{Z}^T \ddot{\mathbf{y}}$ are, respectively,

$$\begin{aligned}
[\mathbf{Z}^T \mathbf{Z} \mathbf{C} \mathbf{b}]_{k\bullet} &= \mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mathbf{C}_{k\bullet} \mathbf{b} = \mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k \\
[\mathbf{Z}^T \ddot{\mathbf{y}}]_{k\bullet} &= [\mathbf{Z}^T]_{k\bullet} \ddot{\mathbf{y}} = (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}}.
\end{aligned}$$

The $(ij)^{\text{th}}$ element of the row vector $(\mathbf{Z} \mathbf{C} \mathbf{b})^T$ is equal to

$$\begin{aligned}
[(\mathbf{Z} \boldsymbol{\mu})^T]_{\bullet(ij)} &= [\boldsymbol{\mu}^T \mathbf{Z}^T]_{\bullet(ij)} = \boldsymbol{\mu}^T (\mathbf{Z}^T)_{\bullet(ij)} \\
&= \boldsymbol{\mu}^T (\mathbf{Z}_{(ij)\bullet})^T
\end{aligned}$$

Therefore, the desired elements are:

$$\begin{aligned}
1. \quad & [(\mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu})(\mathbf{Z} \boldsymbol{\mu})^T]_{k(ij)} = [\mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu}]_{k\bullet} [(\mathbf{Z} \boldsymbol{\mu})^T]_{\bullet(ij)} \\
& = (\mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k) \boldsymbol{\mu}^T (\mathbf{Z}_{(ij)\bullet})^T \\
& = (\mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mu_k) \mathbf{Z}_{(ij)\bullet} \boldsymbol{\mu} \\
& \quad \text{since } \boldsymbol{\mu}^T (\mathbf{Z}_{(ij)\bullet})^T \text{ is a scalar} \\
& = \mu_k \mathbf{1}_n^T \mathbf{Z}_{\bullet k} \mathbf{Z}_{(ij)\bullet} \boldsymbol{\mu} \\
2. \quad & [(\mathbf{Z}^T \ddot{\mathbf{y}})(\mathbf{Z} \boldsymbol{\mu})^T]_{k(ij)} = [\mathbf{Z}^T \ddot{\mathbf{y}}]_{k\bullet} [(\mathbf{Z} \boldsymbol{\mu})^T]_{\bullet(ij)} \\
& = (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}} \boldsymbol{\mu}^T (\mathbf{Z}_{(ij)\bullet})^T \\
& = \ddot{\mathbf{y}}^T \mathbf{Z}_{\bullet k} \mathbf{Z}_{(ij)\bullet} \boldsymbol{\mu}, \\
& \quad \text{since } (\mathbf{Z}_{\bullet k})^T \ddot{\mathbf{y}} \text{ and } \boldsymbol{\mu}^T (\mathbf{Z}_{(ij)\bullet})^T \text{ are both scalars.}
\end{aligned}$$

A formula for calculating $E_{\mathbf{Z}|\mathbf{y}; \psi} [\mathbf{Z}_{\bullet k} \mathbf{Z}_{(ij)\bullet}]$ is given in Proposition 6.3.2, therefore we only need to make the relevant substitutions and look for patterns in the resulting matrix expressions.

$$\begin{aligned}
& E_{\mathbf{Z}|\mathbf{y}; \psi} [(\mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu})(\mathbf{Z} \boldsymbol{\mu})^T]_{k(ij)} \\
& = \mu_k \mathbf{1}_n^T E_{\mathbf{Z}|\mathbf{y}; \psi} [\mathbf{Z}_{\bullet k} \mathbf{Z}_{(ij)\bullet}] \boldsymbol{\mu} \\
& = \mu_k \mathbf{1}_n^T \{ \tilde{\mathbf{Z}}_{\bullet k} \tilde{\mathbf{Z}}_{(ij)\bullet} - \tau_{ik}(y_{ij}; \boldsymbol{\psi}) [\mathbf{I}_n]_{\bullet(ij)} \tilde{\mathbf{Z}}_{(ij)\bullet} + [\mathbf{I}_n]_{\bullet(ij)} [\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{k\bullet} \} \boldsymbol{\mu} \\
& = \mu_k \mathbf{1}_n^T \tilde{\mathbf{Z}}_{\bullet k} \tilde{\mathbf{Z}}_{(ij)\bullet} \boldsymbol{\mu} - \mu_k \mathbf{1}_n^T \tau_{ik}(y_{ij}; \boldsymbol{\psi}) [\mathbf{I}_n]_{\bullet(ij)} \tilde{\mathbf{Z}}_{(ij)\bullet} \boldsymbol{\mu} \\
& \quad + \mu_k \mathbf{1}_n^T [\mathbf{I}_n]_{\bullet(ij)} [\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{k\bullet} \boldsymbol{\mu}.
\end{aligned}$$

Using the identities given in Equations (6.38) to (6.42), we obtain the following.

$$\begin{aligned}
& E_{\mathbf{Z}|\mathbf{y}; \psi} [(\mathbf{Z}^T \mathbf{Z} \boldsymbol{\mu})(\mathbf{Z} \boldsymbol{\mu})^T]_{k(ij)} \\
& = [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu}]_{k\bullet} [\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T]_{\bullet(ij)} - \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \mu_k [\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T]_{\bullet(ij)} + \mu_k^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \\
& = [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu}]_{k\bullet} [\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T]_{\bullet(ij)} - [\text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T]_{k\bullet} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{\bullet(ij)} \\
& \quad + [\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T)]_{k\bullet} [\tilde{\mathbf{Z}}^T]_{\bullet(ij)}
\end{aligned}$$

Therefore,

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\mathbf{Z}\boldsymbol{\mu})(\mathbf{Z}\boldsymbol{\mu})^T]_{k(ij)} \\
&= [\text{diag}(\mathbf{1}_n^T\tilde{\mathbf{Z}})\boldsymbol{\mu}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{k(ij)} - [\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{k(ij)} \\
&\quad + [\text{diag}(\boldsymbol{\mu}^T)\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T]_{k(ij)}
\end{aligned}$$

By the rules of matrix addition and of matrix equality, we have

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\mathbf{Z}\boldsymbol{\mu})(\mathbf{Z}\boldsymbol{\mu})^T] &= \text{diag}(\mathbf{1}_n^T\tilde{\mathbf{Z}})\boldsymbol{\mu}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T - \text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T) \\
&\quad + \text{diag}(\boldsymbol{\mu}^T)\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T.
\end{aligned} \tag{6.66}$$

We proceed in a similar way to evaluate the expectation of $(\mathbf{Z}^T\ddot{\mathbf{y}})(\mathbf{Z}\boldsymbol{\mu})^T$.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\ddot{\mathbf{y}})(\mathbf{Z}\boldsymbol{\mu})^T]_{k(ij)} \\
&= \ddot{\mathbf{y}}^T E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}] \boldsymbol{\mu} \\
&= \ddot{\mathbf{y}}^T \{ \tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(ij)\bullet} - \tau_{ik}(y_{ij};\boldsymbol{\psi})[\mathbf{I}_n]_{\bullet(ij)}\tilde{\mathbf{Z}}_{(ij)\bullet} + [\mathbf{I}_n]_{\bullet(ij)}[\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{k\bullet} \} \boldsymbol{\mu} \\
&= \ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(ij)\bullet} \boldsymbol{\mu} - \ddot{\mathbf{y}}^T \tau_{ik}(y_{ij};\boldsymbol{\psi}) [\mathbf{I}_n]_{\bullet(ij)} \tilde{\mathbf{Z}}_{(ij)\bullet} \boldsymbol{\mu} + \ddot{\mathbf{y}}^T [\mathbf{I}_n]_{\bullet(ij)} [\text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet})]_{k\bullet} \boldsymbol{\mu}.
\end{aligned}$$

Using the identities given in Equations (6.28), (6.39), (6.41), (6.43) and (6.44) we obtain the following simplification.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\ddot{\mathbf{y}})(\mathbf{Z}\boldsymbol{\mu})^T]_{k(ij)} \\
&= [\tilde{\mathbf{Z}}^T\ddot{\mathbf{y}}]_{k\bullet} [\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{\bullet(ij)} - \tau_{ik}(y_{ij};\boldsymbol{\psi}) \ddot{y}_{ij} [\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{\bullet(ij)} + \ddot{y}_{ij} \tau_{ik}(y_{ij};\boldsymbol{\psi}) \mu_k \\
&= [\tilde{\mathbf{Z}}^T\ddot{\mathbf{y}}]_{k\bullet} [\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{\bullet(ij)} - [\tilde{\mathbf{Z}}^T\text{diag}(\ddot{\mathbf{y}}^T)]_{k\bullet} [\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{\bullet(ij)} \\
&\quad + [\text{diag}(\boldsymbol{\mu}^T)]_{k\bullet} [\tilde{\mathbf{Z}}^T\text{diag}(\ddot{\mathbf{y}}^T)]_{\bullet(ij)}
\end{aligned}$$

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\ddot{\mathbf{y}})(\mathbf{Z}\boldsymbol{\mu})^T]_{k(ij)} \\
&= [\tilde{\mathbf{Z}}^T\ddot{\mathbf{y}}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{k(ij)} - [\tilde{\mathbf{Z}}^T\text{diag}(\ddot{\mathbf{y}}^T)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{k(ij)} + [\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T\text{diag}(\ddot{\mathbf{y}}^T)]_{k(ij)}
\end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}^T\ddot{\mathbf{y}})(\mathbf{Z}\boldsymbol{\mu})^T] &= \tilde{\mathbf{Z}}^T\ddot{\mathbf{y}}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T\text{diag}(\ddot{\mathbf{y}}^T)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T) \\
&\quad + \text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T\text{diag}(\ddot{\mathbf{y}}^T).
\end{aligned} \tag{6.67}$$

$$\begin{aligned}
& E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}^*}^T] \\
&= \frac{1}{\sigma^4} \mathbf{C}^T \left[\left(\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T - \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) + \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \right) \right. \\
&\quad - \text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}}) \boldsymbol{\mu} \ddot{\mathbf{y}}^T \\
&\quad - \left(\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}} \boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T - \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) + \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \right) \\
&\quad \left. + (\tilde{\mathbf{Z}}^T \ddot{\mathbf{y}})(\ddot{\mathbf{y}})^T \right] \mathbf{X}_2
\end{aligned}$$

Simplifying the above expression yields the result of Proposition 6.4.7. \square

Proposition 6.4.8.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\mathbf{b}^*}^T] &= (E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}^*}]) (E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}^*}])^T \\
&\quad + \frac{1}{\sigma^4} \mathbf{X}_2^T \left[\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T) - \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \right] \mathbf{X}_2.
\end{aligned}$$

Proof of Proposition 6.4.8.

$$\begin{aligned}
\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\mathbf{b}^*}^T &= \frac{1}{\sigma^4} \mathbf{X}_2^T (\mathbf{Z} \mathbf{C} \mathbf{b} - \ddot{\mathbf{y}})(\mathbf{Z} \mathbf{C} \mathbf{b} - \ddot{\mathbf{y}})^T \mathbf{X}_2 \\
&= \frac{1}{\sigma^4} \mathbf{X}_2^T \left[\mathbf{Z} \boldsymbol{\mu} (\mathbf{Z} \boldsymbol{\mu})^T - \mathbf{Z} \boldsymbol{\mu} \ddot{\mathbf{y}}^T - \ddot{\mathbf{y}} (\mathbf{Z} \boldsymbol{\mu})^T + \ddot{\mathbf{y}} \ddot{\mathbf{y}}^T \right] \mathbf{X}_2 \tag{6.68}
\end{aligned}$$

Therefore,

$$E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\mathbf{b}^*}^T] = \frac{1}{\sigma^4} \mathbf{X}_2^T \left[E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z} \boldsymbol{\mu} (\mathbf{Z} \boldsymbol{\mu})^T] - \tilde{\mathbf{Z}} \boldsymbol{\mu} \ddot{\mathbf{y}}^T - \ddot{\mathbf{y}} (\tilde{\mathbf{Z}} \boldsymbol{\mu})^T + \ddot{\mathbf{y}} \ddot{\mathbf{y}}^T \right] \mathbf{X}_2$$

and so we need to calculate $E_{\mathbf{Z}|\mathbf{y}; \psi}[\mathbf{Z} \boldsymbol{\mu} (\mathbf{Z} \boldsymbol{\mu})^T]$.

The matrix $\mathbf{Z} \boldsymbol{\mu} (\mathbf{Z} \boldsymbol{\mu})^T = \mathbf{Z} \boldsymbol{\mu} \boldsymbol{\mu}^T \mathbf{Z}^T$ is $n \times n$ and the element in row (ij) and column $(i'j')$ is given by:

$$\begin{aligned}
[\mathbf{Z}\boldsymbol{\mu}\boldsymbol{\mu}^T\mathbf{Z}^T]_{(ij)(i'j')} &= [\mathbf{Z}\boldsymbol{\mu}]_{(ij)\bullet} [\boldsymbol{\mu}^T\mathbf{Z}^T]_{\bullet(i'j')} \\
&= \mathbf{Z}_{(ij)\bullet}\boldsymbol{\mu}\boldsymbol{\mu}^T(\mathbf{Z}^T)_{\bullet(i'j')} \\
&= \mathbf{Z}_{(ij)\bullet}\boldsymbol{\mu}\boldsymbol{\mu}^T(\mathbf{Z}_{(i'j')\bullet})^T \\
&= \boldsymbol{\mu}^T(\mathbf{Z}_{(ij)\bullet})^T\mathbf{Z}_{(i'j')\bullet}\boldsymbol{\mu},
\end{aligned}$$

since $\mathbf{Z}_{(ij)\bullet}\boldsymbol{\mu}$ and $\boldsymbol{\mu}^T(\mathbf{Z}_{(i'j')\bullet})^T$ are both scalars

A formula for calculating $E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}_{(ij)\bullet})^T\mathbf{Z}_{(i'j')\bullet}]$ is given in Proposition 6.3.3, therefore we only need to make the relevant substitutions and look for patterns in the resulting matrix expressions.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}\boldsymbol{\mu}\boldsymbol{\mu}^T\mathbf{Z}^T]_{(ij)(i'j')} &= \boldsymbol{\mu}^T E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}_{(ij)\bullet})^T\mathbf{Z}_{(i'j')\bullet}]\boldsymbol{\mu} \\
&= \boldsymbol{\mu}^T \{ (1 - \delta_{(ij)(i'j')}) (\tilde{\mathbf{Z}}_{(ij)\bullet})^T \tilde{\mathbf{Z}}_{(i'j')\bullet} + \delta_{(ij)(i'j')} \text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet}) \} \boldsymbol{\mu} \\
&= \boldsymbol{\mu}^T (\tilde{\mathbf{Z}}_{(ij)\bullet})^T \tilde{\mathbf{Z}}_{(i'j')\bullet} \boldsymbol{\mu} - \delta_{(ij)(i'j')} \boldsymbol{\mu}^T (\tilde{\mathbf{Z}}_{(ij)\bullet})^T \tilde{\mathbf{Z}}_{(i'j')\bullet} \boldsymbol{\mu} + \delta_{(ij)(i'j')} \boldsymbol{\mu}^T \text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet}) \boldsymbol{\mu}
\end{aligned}$$

Using the identities given in Equations (6.39), (6.45), and (6.46) we obtain the next three simplifications.

$$\boldsymbol{\mu}^T (\tilde{\mathbf{Z}}_{(ij)\bullet})^T \tilde{\mathbf{Z}}_{(i'j')\bullet} \boldsymbol{\mu} = [\tilde{\mathbf{Z}}\boldsymbol{\mu}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{(ij)(i'j')},$$

$$\begin{aligned}
\delta_{(ij)(i'j')} \boldsymbol{\mu}^T (\tilde{\mathbf{Z}}_{(ij)\bullet})^T \tilde{\mathbf{Z}}_{(i'j')\bullet} \boldsymbol{\mu} &= [\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{(ij)\bullet} [\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{\bullet(i'j')} \\
&= [\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{(ij)(i'j')}
\end{aligned}$$

and

$$\delta_{(ij)(i'j')} \boldsymbol{\mu}^T \text{diag}(\tilde{\mathbf{Z}}_{(ij)\bullet}) \boldsymbol{\mu} = [\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T)]_{(ij)(i'j')}.$$

Hence,

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}\boldsymbol{\mu}\boldsymbol{\mu}^T\mathbf{Z}^T]_{(ij)(i'j')} \\
&= [\tilde{\mathbf{Z}}\boldsymbol{\mu}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{(ij)(i'j')} - [\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{(ij)(i'j')} \\
&\quad + [\text{diag}(\boldsymbol{\mu}^T\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T)]_{(ij)(i'j')}
\end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}\boldsymbol{\mu}\boldsymbol{\mu}^T\mathbf{Z}^T] &= \tilde{\mathbf{Z}}\boldsymbol{\mu}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T - \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T) \\
&\quad + \text{diag}(\boldsymbol{\mu}^T\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T) \tag{6.69}
\end{aligned}$$

and so:

$$\begin{aligned}
&E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{\mathbf{b}^*}^T] \\
&= \frac{1}{\sigma^4}\mathbf{X}_2^T \left[\left(\tilde{\mathbf{Z}}\boldsymbol{\mu}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T - \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T) + \text{diag}(\boldsymbol{\mu}^T\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T) \right) \right. \\
&\quad \left. - \tilde{\mathbf{Z}}\boldsymbol{\mu}\ddot{\mathbf{y}}^T - \ddot{\mathbf{y}}(\tilde{\mathbf{Z}}\boldsymbol{\mu})^T + \ddot{\mathbf{y}}\ddot{\mathbf{y}}^T \right] \mathbf{X}_2 \\
&= (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}])^T \\
&\quad + \frac{1}{\sigma^4}\mathbf{X}_2^T \left[\text{diag}(\boldsymbol{\mu}^T\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T) - \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T) \right] \mathbf{X}_2.
\end{aligned}$$

This proves Proposition 6.4.8. \square

Proposition 6.4.9.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{(\sigma^2)}^T] &= (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}])^T \\
&\quad + \frac{1}{2\sigma^6}\mathbf{X}_2^T \left[2\text{diag}(\boldsymbol{\mu}^T\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T)\ddot{\mathbf{y}} \right. \\
&\quad - 2\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\ddot{\mathbf{y}} \\
&\quad \left. - \tilde{\mathbf{Z}}\text{diag}(\boldsymbol{\mu}^T)\text{diag}(\boldsymbol{\mu}^T)\boldsymbol{\mu} + \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\tilde{\mathbf{Z}}\text{diag}(\boldsymbol{\mu}^T)\boldsymbol{\mu} \right]
\end{aligned}$$

Proof of Proposition 6.4.9.

$$\begin{aligned}
\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{(\sigma^2)}^T &= \frac{n}{2\sigma^4}\mathbf{X}_2(\mathbf{Z}\boldsymbol{\mu} - \ddot{\mathbf{y}}) + \frac{1}{2\sigma^6}\mathbf{X}_2^T\ddot{\mathbf{y}} \left(\ddot{\mathbf{y}}^T\ddot{\mathbf{y}} - 2(\mathbf{Z}\boldsymbol{\mu})^T\ddot{\mathbf{y}} + \boldsymbol{\mu}^T\mathbf{Z}^T\mathbf{Z}\boldsymbol{\mu} \right) \\
&\quad - \frac{1}{2\sigma^6}\mathbf{X}_2^T \left(\mathbf{Z}\boldsymbol{\mu}\ddot{\mathbf{y}}^T\ddot{\mathbf{y}} - 2(\mathbf{Z}\boldsymbol{\mu})(\mathbf{Z}\boldsymbol{\mu})^T\ddot{\mathbf{y}} + \mathbf{Z}\boldsymbol{\mu}(\mathbf{Z}^T\mathbf{Z}\boldsymbol{\mu})^T\boldsymbol{\mu} \right) \tag{6.70}
\end{aligned}$$

The expectations needed to calculate $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{(\sigma^2)}^T]$ are known from Equations (5.48) to (5.55) and from Equations (6.66) and (6.69). It only remains to make the substitutions and simplify the resulting expression.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{(\sigma^2)}^T] &= \frac{n}{2\sigma^4}\mathbf{X}_2(\tilde{\mathbf{Z}}\boldsymbol{\mu} - \ddot{\mathbf{y}}) + \frac{1}{2\sigma^6}\mathbf{X}_2^T\ddot{\mathbf{y}}\left(\ddot{\mathbf{y}}^T\ddot{\mathbf{y}} - 2(\tilde{\mathbf{Z}}\boldsymbol{\mu})^T\ddot{\mathbf{y}} + \boldsymbol{\mu}^T\text{diag}(\mathbf{1}_n^T\tilde{\mathbf{Z}})\boldsymbol{\mu}\right) \\
&\quad - \frac{1}{2\sigma^6}\mathbf{X}_2^T\left[\tilde{\mathbf{Z}}\boldsymbol{\mu}\ddot{\mathbf{y}}^T\ddot{\mathbf{y}} \right. \\
&\quad \left. - 2\left(\tilde{\mathbf{Z}}\boldsymbol{\mu}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T - \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T) + \text{diag}(\boldsymbol{\mu}^T\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T)\right)\ddot{\mathbf{y}} \right. \\
&\quad \left. + \left(\tilde{\mathbf{Z}}\boldsymbol{\mu}\boldsymbol{\mu}^T\text{diag}(\mathbf{1}_n^T\tilde{\mathbf{Z}}) - \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\tilde{\mathbf{Z}}\text{diag}(\boldsymbol{\mu}^T) + \tilde{\mathbf{Z}}\text{diag}(\boldsymbol{\mu}^T)\text{diag}(\boldsymbol{\mu}^T)\right)\boldsymbol{\mu}\right]
\end{aligned}$$

After simplifying the above result, we obtain the result of Proposition 6.4.9. \square

Proposition 6.4.10.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{\phi}^T] &= (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\phi}])^T \\
&\quad + \frac{1}{\sigma^2}\mathbf{X}_2^T \left[\sum_{i=1}^s \left[\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\tilde{\mathbf{Z}} \right. \right. \\
&\quad \left. \left. - \text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\tilde{\mathbf{Z}}\text{diag}(\boldsymbol{\mu}^T) \right] \left(\frac{\partial}{\partial\phi} \mathbf{h}_i(\phi) \right) \right].
\end{aligned}$$

Proof of Proposition 6.4.10.

$$\begin{aligned}
\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{\phi}^T &= -\frac{1}{\sigma^2}\mathbf{X}_2^T(\mathbf{Z}\mathbf{C}\mathbf{b} - \ddot{\mathbf{y}}) \left[\sum_{i=1}^s \mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\mathbf{Z} \left(\frac{\partial}{\partial\phi} \mathbf{h}_i(\phi) \right) \right] \\
&= \frac{1}{\sigma^2}\mathbf{X}_2^T\ddot{\mathbf{y}} \left[\sum_{i=1}^s \mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\mathbf{Z} \left(\frac{\partial}{\partial\phi} \mathbf{h}_i(\phi) \right) \right] \\
&\quad - \frac{1}{\sigma^2}\mathbf{X}_2^T \left[\sum_{i=1}^s (\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\mathbf{Z} \left(\frac{\partial}{\partial\phi} \mathbf{h}_i(\phi) \right) \right]
\end{aligned} \tag{6.71}$$

Therefore,

$$E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{\phi}^T] = \frac{1}{\sigma^2}\mathbf{X}_2^T\ddot{\mathbf{y}}\left[\sum_{i=1}^s\mathbf{1}_{n_i}^T\Delta_i\tilde{\mathbf{Z}}\left(\frac{\partial}{\partial\phi}\mathbf{h}_i(\phi)\right)\right] \\ - \frac{1}{\sigma^2}\mathbf{X}_2^T\left[\sum_{i=1}^s E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}]\left(\frac{\partial}{\partial\phi}\mathbf{h}_i(\phi)\right)\right]$$

and so we need to calculate $E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}]$.

The element in row $(i'j')$ and column k of $(\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}$ is given by:

$$\begin{aligned} [(\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}]_{(i'j')k} &= [\mathbf{Z}\boldsymbol{\mu}]_{(i'j')\bullet}[\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}]_{\bullet k} \\ &= \mathbf{Z}_{(i'j')\bullet}\boldsymbol{\mu}\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}_{\bullet k} \\ &= \boldsymbol{\mu}^T(\mathbf{Z}_{(i'j')\bullet})^T(\mathbf{Z}_{\bullet k})^T\Delta_i^T\mathbf{1}_{n_i} \\ &\quad \text{since } \mathbf{Z}_{(i'j')\bullet}\boldsymbol{\mu} \text{ and } \mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}_{\bullet k} \text{ are both scalars} \\ &\quad \text{and so are both symmetric} \\ &= \boldsymbol{\mu}^T(\mathbf{Z}_{\bullet k}\mathbf{Z}_{(i'j')\bullet})^T\Delta_i^T\mathbf{1}_{n_i} \\ &= \mathbf{1}_{n_i}^T\Delta_i(\mathbf{Z}_{\bullet k}\mathbf{Z}_{(i'j')\bullet})\boldsymbol{\mu}, \text{ since we have a scalar.} \end{aligned}$$

A formula for calculating $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(ij)\bullet}]$ is given in Proposition 6.3.2, therefore we only need to make the relevant substitutions and look for patterns in the resulting matrix expressions.

$$\begin{aligned} E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\Delta_i\mathbf{Z}]_{(i'j')k} &= \mathbf{1}_{n_i}^T\Delta_i E_{\mathbf{Z}|\mathbf{y};\psi}[\mathbf{Z}_{\bullet k}\mathbf{Z}_{(i'j')\bullet}]\boldsymbol{\mu} \\ &= \mathbf{1}_{n_i}^T\Delta_i\left\{\tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(i'j')\bullet} - \tau_{i'k}(y_{i'j'};\boldsymbol{\psi})[\mathbf{I}_n]_{\bullet(i'j')}\tilde{\mathbf{Z}}_{(i'j')\bullet} + [\mathbf{I}_n]_{\bullet(i'j')}\text{diag}(\tilde{\mathbf{Z}}_{(i'j')\bullet})_{k\bullet}\right\}\boldsymbol{\mu} \\ &= \mathbf{1}_{n_i}^T\Delta_i\tilde{\mathbf{Z}}_{\bullet k}\tilde{\mathbf{Z}}_{(i'j')\bullet}\boldsymbol{\mu} - \mathbf{1}_{n_i}^T\Delta_i\tau_{i'k}(y_{i'j'};\boldsymbol{\psi})[\mathbf{I}_n]_{\bullet(i'j')}\tilde{\mathbf{Z}}_{(i'j')\bullet}\boldsymbol{\mu} \\ &\quad + \mathbf{1}_{n_i}^T\Delta_i[\mathbf{I}_n]_{\bullet(i'j')}\text{diag}(\tilde{\mathbf{Z}}_{(i'j')\bullet})_{k\bullet}\boldsymbol{\mu}. \end{aligned}$$

Using the identities given in Equations (6.31), (6.39), (6.41), (6.47) and (6.48) we obtain the following simplification.

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\mathbf{Z}]_{(i'j')k} &= [\tilde{\mathbf{Z}}^T\boldsymbol{\Delta}_i^T\mathbf{1}_{n_i}]_{k\bullet}[\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{\bullet(i'j')} - [\tilde{\mathbf{Z}}^T\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)]_{k(i'j')}[\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{\bullet(i'j')} \\
&\quad + \tau_{i'k}(y_{i'j'}; \boldsymbol{\psi})\mu_k[\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i]_{\bullet(i'j')} \\
&= [\tilde{\mathbf{Z}}^T\boldsymbol{\Delta}_i^T\mathbf{1}_{n_i}]_{k\bullet}[\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{\bullet(i'j')} - [\tilde{\mathbf{Z}}^T\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)]_{k\bullet}[\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{\bullet(i'j')} \\
&\quad + [\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T]_{k\bullet}[\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)]_{\bullet(i'j')}
\end{aligned}$$

Hence,

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\mathbf{Z}]_{(i'j')k} &= [\tilde{\mathbf{Z}}^T\boldsymbol{\Delta}_i^T\mathbf{1}_{n_i}\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T]_{k(i'j')} - [\tilde{\mathbf{Z}}^T\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)]_{k(i'j')} \\
&\quad + [\text{diag}(\boldsymbol{\mu}^T)\tilde{\mathbf{Z}}^T\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)]_{k(i'j')} \\
&= [\tilde{\mathbf{Z}}\boldsymbol{\mu}\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\tilde{\mathbf{Z}}]_{(i'j')k} - [\text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\tilde{\mathbf{Z}}]_{(i'j')k} \\
&\quad + [\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\tilde{\mathbf{Z}}\text{diag}(\boldsymbol{\mu}^T)]_{(i'j')k}.
\end{aligned}$$

Therefore, by the rules of matrix addition and of matrix equality, we have

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[(\mathbf{Z}\boldsymbol{\mu})\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\mathbf{Z}] &= \tilde{\mathbf{Z}}\boldsymbol{\mu}\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\tilde{\mathbf{Z}} - \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\tilde{\mathbf{Z}} \\
&\quad + \text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\tilde{\mathbf{Z}}\text{diag}(\boldsymbol{\mu}^T)
\end{aligned} \tag{6.72}$$

Substituting this result into the expression for $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{\boldsymbol{\phi}}^T]$, we obtain:

$$\begin{aligned}
E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}^*}\mathcal{U}_{\boldsymbol{\phi}}^T] &= \frac{1}{\sigma^2}\mathbf{X}_2^T\ddot{\mathbf{y}}\left[\sum_{i=1}^s\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\tilde{\mathbf{Z}}\left(\frac{\partial}{\partial\boldsymbol{\phi}}\mathbf{h}_i(\boldsymbol{\phi})\right)\right] \\
&\quad - \frac{1}{\sigma^2}\mathbf{X}_2^T\left[\sum_{i=1}^s\left(\tilde{\mathbf{Z}}\boldsymbol{\mu}\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i\tilde{\mathbf{Z}} - \text{diag}(\boldsymbol{\mu}^T\tilde{\mathbf{Z}}^T)\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\tilde{\mathbf{Z}}\right)\left(\frac{\partial}{\partial\boldsymbol{\phi}}\mathbf{h}_i(\boldsymbol{\phi})\right)\right] \\
&\quad - \frac{1}{\sigma^2}\mathbf{X}_2^T\left[\sum_{i=1}^s\text{diag}(\mathbf{1}_{n_i}^T\boldsymbol{\Delta}_i)\tilde{\mathbf{Z}}\text{diag}(\boldsymbol{\mu}^T)\left(\frac{\partial}{\partial\boldsymbol{\phi}}\mathbf{h}_i(\boldsymbol{\phi})\right)\right]
\end{aligned}$$

Simplifying the above expression yields the result of Proposition 6.4.10. \square

6.5 The Conditional Observed Information Matrix

In Chapter 5, Equation (5.84), the conditional observed information was partitioned into blocks, and formulae for calculating the blocks were given in Equations (5.85) to (5.94). This section shows how the foregoing results (Equations (6.1) to (6.4) and Propositions 6.4.1 to 6.4.10) are used to prove Equations (5.85) to (5.94). Essentially, this section outlines the set of substitutions that must be made in order to derive the required formulae.

Proof of Equation (5.85).

$$\begin{aligned}\mathcal{I}_{\mathbf{bb}}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{bb}}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}, \mathcal{U}_{\mathbf{b}}] \\ &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{bb}}] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}}^T] + (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}]) (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}])^T\end{aligned}$$

The expectation $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}]$ is known from Equation (6.1). For $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}}^T]$, substitute the result of Proposition 6.4.1. The result follows. \square

Proof of Equation (5.86).

$$\begin{aligned}\mathcal{I}_{(\sigma^2)(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{(\sigma^2)(\sigma^2)}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}, \mathcal{U}_{(\sigma^2)}] \\ &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{(\sigma^2)(\sigma^2)}] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)} \mathcal{U}_{(\sigma^2)}^T] \\ &\quad + (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}]) (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}])^T\end{aligned}$$

The expectation $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}]$ is known from Equation (6.3). For $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)} \mathcal{U}_{(\sigma^2)}^T]$, substitute the result of Proposition 6.4.2. \square

Proof of Equation (5.87).

$$\begin{aligned}\mathcal{I}_{\phi\phi}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\phi\phi}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\phi}, \mathcal{U}_{\phi}] \\ &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\phi\phi}] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\phi} \mathcal{U}_{\phi}^T] + (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\phi}]) (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\phi}])^T\end{aligned}$$

The expectation $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi]$ is known from Equation (6.4). For $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi \mathcal{U}_\phi^T]$ substitute the result of Proposition 6.4.3. \square

Proof of Equation (5.88).

$$\begin{aligned}\mathcal{I}_{\mathbf{b}(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}(\sigma^2)}] - \text{cov}_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}}, \mathcal{U}_{(\sigma^2)}] \\ &= E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}(\sigma^2)}] - E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{(\sigma^2)}^T] + (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}])^T\end{aligned}$$

The expectations $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}}]$ and $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}]$ are known from Equations (6.1) and (6.3), respectively. For $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{(\sigma^2)}^T]$, we substitute the result of Proposition 6.4.4. \square

Proof of Equation (5.89).

$$\begin{aligned}\mathcal{I}_{\mathbf{b}\phi}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}\phi}] - \text{cov}_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}}, \mathcal{U}_\phi] \\ &= E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{\mathbf{b}\phi}] - E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_\phi^T] + (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi])^T\end{aligned}$$

The expectations $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}}]$ and $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi]$ are known from Equations (6.1) and (6.4), respectively. For $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_\phi^T]$, we substitute the result of Proposition 6.4.5. \square

Proof of Equation (5.90).

$$\begin{aligned}\mathcal{I}_{(\sigma^2)\phi}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{(\sigma^2)\phi}] - \text{cov}_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}, \mathcal{U}_\phi] \\ &= E_{\mathbf{Z}|\mathbf{y};\psi}[-\mathcal{U}_{(\sigma^2)\phi}] - E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)} \mathcal{U}_\phi^T] + (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}]) (E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi])^T\end{aligned}$$

The expectations $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)}]$ and $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_\phi]$ are known from Equations (6.3) and (6.4), respectively. We substitute the value of $E_{\mathbf{Z}|\mathbf{y};\psi}[\mathcal{U}_{(\sigma^2)} \mathcal{U}_\phi^T]$ from Proposition 6.4.6. \square

Proof of Equation (5.91).

$$\begin{aligned}\mathcal{I}_{\mathbf{b}\mathbf{b}^*}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}\mathbf{b}^*}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}, \mathcal{U}_{\mathbf{b}^*}] \\ &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}\mathbf{b}^*}] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}^*}^T] + (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}]) (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}])^T\end{aligned}$$

The expectations $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}}]$ and $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}]$ are known from Equations (6.1) and (6.2). For $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}} \mathcal{U}_{\mathbf{b}^*}^T]$, we use the results of Proposition 6.4.7. \square

Proof of Equation (5.92).

$$\begin{aligned}\mathcal{I}_{\mathbf{b}^*\mathbf{b}^*}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*\mathbf{b}^*}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}, \mathcal{U}_{\mathbf{b}^*}] \\ &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*\mathbf{b}^*}] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\mathbf{b}^*}^T] + (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}]) (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}])^T\end{aligned}$$

The conditional expectation $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}]$ is known from Equation (6.2). We substitute the value of $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\mathbf{b}^*}^T]$ from Proposition 6.4.8. \square

Proof of Equation (5.93).

$$\begin{aligned}\mathcal{I}_{\mathbf{b}^*(\sigma^2)}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*(\sigma^2)}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}, \mathcal{U}_{(\sigma^2)}] \\ &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*(\sigma^2)}] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{(\sigma^2)}^T] + (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}]) (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}])^T\end{aligned}$$

The expectations $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}]$ and $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{(\sigma^2)}]$ are known from Equations (6.2) and (6.3), respectively. For the expectation $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{(\sigma^2)}^T]$, we substitute the results of Proposition 6.4.9. \square

Proof of Equation 5.94.

$$\begin{aligned}\mathcal{I}_{\mathbf{b}^*\phi}(\boldsymbol{\psi}; \mathbf{y}) &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*\phi}] - \text{cov}_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}, \mathcal{U}_{\phi}] \\ &= E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[-\mathcal{U}_{\mathbf{b}^*\phi}] - E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\phi}^T] + (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}]) (E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\phi}])^T\end{aligned}$$

The conditional expectations $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*}]$ and $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\phi}]$ are known from Equations (6.2) and (6.4). For the expectation, $E_{\mathbf{Z}|\mathbf{y}; \boldsymbol{\psi}}[\mathcal{U}_{\mathbf{b}^*} \mathcal{U}_{\phi}^T]$, we substitute the results of Proposition 6.4.10. \square

6.6 Intermediate Results Involving Integrals

In order to calculate the Fisher information matrix, we take the expectation, over the distribution of \mathbf{y} , of the conditional information matrix. This section presents some intermediate integrals (see Table 6.3) that are used to facilitate this calculation.

Table 6.3: List of integrals used for calculating of the Fisher information matrix.

Integral (or expectation)	Proposition that reveals a formula for this integral
$E_{\mathbf{y}; \boldsymbol{\psi}}[\tilde{z}_{ijk}]$	Proposition 6.6.1
$E_{\mathbf{y}; \boldsymbol{\psi}}[\tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})]$	Proposition 6.6.2
$E_{\mathbf{y}; \boldsymbol{\psi}}[\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi})]$	Proposition 6.6.3
$E_{\mathbf{y}; \boldsymbol{\psi}}[\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})]$	Proposition 6.6.4
$E_{\mathbf{y}; \boldsymbol{\psi}}[\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi})]$	Proposition 6.6.5
$E_{\mathbf{y}; \boldsymbol{\psi}}[\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})]$	Proposition 6.6.6

Proposition 6.6.1.

$$E_{\mathbf{y}; \boldsymbol{\psi}}[\tilde{z}_{ijk}] = w_{ik}$$

Proof of Proposition 6.6.1. From Equations (5.32), (5.33), (5.47) and (5.48) we know that the element in column (ij) and row k of $\tilde{\mathbf{Z}}$ is given by

$$\tilde{z}_{ijk} = \tau_{ik}(y_{ij}; \boldsymbol{\psi}) = \frac{w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi})}{f(y_{ij}; \boldsymbol{\psi})}$$

$$\begin{aligned}
 E_{\mathbf{y}; \boldsymbol{\psi}}[\tilde{z}_{ijk}] &= \int \tau_{ik}(y_{ij}; \boldsymbol{\psi}) f(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\
 &= \int \frac{w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi})}{f(y_{ij}; \boldsymbol{\psi})} f(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\
 &= w_{ik} \int f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij} = w_{ik}
 \end{aligned}$$

□

Proposition 6.6.2.

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] = \begin{cases} 0 & \text{if } k \neq k' \\ w_{ik} & \text{if } k = k'. \end{cases}$$

Proof of Proposition 6.6.2. We will examine the cases $k \neq k'$, and $k = k'$, separately.

Suppose that $k \neq k'$. Then

$$\begin{aligned} E_{\mathbf{y}; \boldsymbol{\psi}} [\tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] &= \int \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) f(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\ &= \int \frac{w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi})}{f(y_{ij}; \boldsymbol{\psi})} w_{ik'} f_{ik'}(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\ &= w_{ik'} \frac{w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi})}{f(y_{ij}; \boldsymbol{\psi})} - \int w_{ik'} \frac{d}{dy_{ij}} \left(\frac{w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi})}{f(y_{ij}; \boldsymbol{\psi})} \right) dy_{ij} \\ &= w_{ik'} \frac{w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi})}{f(y_{ij}; \boldsymbol{\psi})} - w_{ik'} \int \frac{d}{dy_{ij}} \left(\frac{w_{ik} f_{ik}(y_{ij}; \boldsymbol{\psi})}{f(y_{ij}; \boldsymbol{\psi})} \right) dy_{ij} \\ &\quad \text{since } w_{ik'} \text{ is constant} \\ &= 0. \end{aligned}$$

Suppose that $k = k'$. Then, using the fact that $\sum_k \tau_{ik}(y_{ij}; \boldsymbol{\psi}) = 1$, we have the following result.

$$\begin{aligned} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) &= (\tau_{ik}(y_{ij}; \boldsymbol{\psi}))^2 \\ &= \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \left[1 - \sum_{k \neq k'} \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) \right] \\ &= \tau_{ik}(y_{ij}; \boldsymbol{\psi}) - \sum_{k \neq k'} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}). \end{aligned}$$

Therefore, if $k = k'$, we have that

$$\begin{aligned} E_{\mathbf{y}; \boldsymbol{\psi}} [\tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] &= E_{\mathbf{y}; \boldsymbol{\psi}} [\tau_{ik}(y_{ij}; \boldsymbol{\psi})] - \sum_{k \neq k'} E_{\mathbf{y}; \boldsymbol{\psi}} [\tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] \\ &= w_{ik} - 0 \\ &= w_{ik}. \end{aligned}$$

□

Proposition 6.6.3.

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi})] = w_{ik} \mu_k$$

Proof of Proposition 6.6.3.

$$\begin{aligned}
E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi})] &= \int \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) f(y_{ij}; \boldsymbol{\psi}) dy_{ij}, \\
&= \int (y_{ij} - \mu_{ij}^*) \tau_{ik}(y_{ij}; \boldsymbol{\psi}) f(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\
&= w_{ik} \int (y_{ij} - \mu_{ij}^*) f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\
&= w_{ik} \int y_{ij} f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij} - w_{ik} \mu_{ij}^* \int f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\
&= w_{ik} (\mu_{ij}^* + \mu_k) - w_{ik} \mu_{ij}^* (1) \\
&= w_{ik} \mu_k.
\end{aligned}$$

□

Proposition 6.6.4.

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] = \begin{cases} 0 & \text{if } k \neq k' \\ w_{ik} \mu_k & \text{if } k = k'. \end{cases}$$

Proof of Proposition 6.6.4. We will examine the cases $k \neq k'$, and $k = k'$, separately.

Suppose that $k \neq k'$. Then

$$\begin{aligned}
E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] &= \int \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) f(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\
&= w_{ik} \mu_k \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) - w_{ik} \mu_k \int \frac{d}{dy_{ij}} \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) dy_{ij}, \\
&\quad \text{using Proposition 6.6.3 and integration by parts} \\
&= 0.
\end{aligned}$$

Suppose that $k = k'$. Then, using the fact that $\sum_k \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) = 1$, we have the following result.

$$\begin{aligned} \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) &= \ddot{y}_{ij} (\tau_{ik}(y_{ij}; \boldsymbol{\psi}))^2 \\ &= \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \left[1 - \sum_{k \neq k'} \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) \right] \\ &= \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) - \sum_{k \neq k'} \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}). \end{aligned}$$

Therefore, if $k = k'$, we have that

$$\begin{aligned} E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] \\ &= E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi})] - \sum_{k \neq k'} E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] \\ &= w_{ik} \mu_k, \text{ using the previous result and Proposition 6.6.3.} \end{aligned}$$

□

Proposition 6.6.5.

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi})] = w_{ik} (\sigma^2 + \mu_k^2).$$

Proof of Proposition 6.6.5.

$$\begin{aligned} E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi})] &= \int \ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) f(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\ &= w_{ik} \int (y_{ij} - \mu_{ij}^*)^2 f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij}, \\ &= w_{ik} \int (y_{ij}^2 - 2 \mu_{ij}^* y_{ij} + (\mu_{ij}^*)^2) f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij}, \\ &= w_{ik} \int y_{ij}^2 f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij} - 2 w_{ik} \mu_{ij}^* \int y_{ij} f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\ &\quad + w_{ik} (\mu_{ij}^*)^2 \int f_{ik}(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\ &= w_{ik} (\sigma^2 + (\mu_{ij}^*)^2 + 2 \mu_k \mu_{ij}^* + \mu_k^2) - 2 w_{ik} \mu_{ij}^* (\mu_{ij}^* + \mu_k) + w_{ik} (\mu_{ij}^*)^2 (1) \\ &= w_{ik} (\sigma^2 + \mu_k^2) \end{aligned}$$

□

Proposition 6.6.6.

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] = \begin{cases} 0 & \text{if } k \neq k' \\ w_{ik} (\sigma^2 + \mu_k^2) & \text{if } k = k'. \end{cases}$$

Proof of Proposition 6.6.6. We will examine the cases $k \neq k'$, and $k = k'$, separately.

Suppose that $k \neq k'$. Then

$$\begin{aligned} & E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] \\ &= \int \ddot{y}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) f(y_{ij}; \boldsymbol{\psi}) dy_{ij} \\ &= w_{ik} (\sigma^2 + \mu_k^2) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) - w_{ik} (\sigma^2 + \mu_k^2) \int \frac{d}{dy_{ij}} \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) dy_{ij}, \\ & \quad \text{using Proposition 6.6.5 and integration by parts} \\ &= 0. \end{aligned}$$

Suppose that $k = k'$. Then, using the fact that $\sum_k \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) = 1$, we have the following result.

$$\begin{aligned} \ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) &= \ddot{y}_{ij}^2 (\tau_{ik}(y_{ij}; \boldsymbol{\psi}))^2 \\ &= \ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \left[1 - \sum_{k \neq k'} \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) \right] \\ &= \ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) - \sum_{k \neq k'} \ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}). \end{aligned}$$

Therefore, if $k = k'$, we have that

$$\begin{aligned} & E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] \\ &= E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi})] - \sum_{k \neq k'} E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] \\ &= w_{ik} (\sigma^2 + \mu_k^2), \text{ using the previous result and Proposition 6.6.5.} \end{aligned}$$

□

6.7 The Fisher Information Matrix

The Fisher (or expected) information is given by

$$\mathcal{I}(\boldsymbol{\psi}) = E_{\mathbf{y}; \boldsymbol{\psi}} [\mathcal{I}(\boldsymbol{\psi}; \mathbf{y})].$$

To evaluate the expectation (over the distribution of \mathbf{y}) of the blocks of $\mathcal{I}(\boldsymbol{\psi}; \mathbf{y})$ we first prove that Equations (6.73) to (6.94) are true. Then the Fisher information matrix formula given in Equation (5.97) follows by direct substitution.

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}] = \bar{\mathbf{Z}} \text{ (as given in Equation (5.95))} \quad (6.73)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\mathbf{1}_n^T \tilde{\mathbf{Z}})] = \text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}}) \quad (6.74)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \Delta_i^T \tilde{\mathbf{Z}}_i)] = \text{diag}(\mathbf{1}_{n_i}^T \bar{\mathbf{Z}}_i \Delta_i^T \bar{\mathbf{Z}}_i) \quad (6.75)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\mathbf{1}_{n_i}^T \tilde{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T))] = \text{diag}(\mathbf{1}_{n_i}^T \bar{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) \quad (6.76)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{\mathbf{y}}] = \bar{\mathbf{Z}} \boldsymbol{\mu} \quad (6.77)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}}] = \text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}}) \quad (6.78)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i] = \text{diag}(\mathbf{1}_{n_i}^T \bar{\mathbf{Z}}_i) \quad (6.79)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}_i^T \Delta_i \Delta_i^T \tilde{\mathbf{Z}}_i] = \text{diag}(\mathbf{1}_{n_i}^T \bar{\mathbf{Z}}_i \Delta_i^T \bar{\mathbf{Z}}_i) \quad (6.80)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}}] = \mathbf{1}_n^T \bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \quad (6.81)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\ddot{\mathbf{y}}^T \tilde{\mathbf{Z}})] = \text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T)) \quad (6.82)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\ddot{\mathbf{y}}^T \Delta_i^T \tilde{\mathbf{Z}}_i)] = \text{diag}(\mathbf{1}_{n_i}^T \bar{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) \quad (6.83)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}] = \text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T)) \quad (6.84)$$

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \Delta_i^T \tilde{\mathbf{Z}}_i] = \text{diag}(\mathbf{1}_{n_i}^T \bar{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)) \quad (6.85)$$

$$E_{\mathbf{y}; \psi} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T)] = \text{diag}(\boldsymbol{\mu}^T) \bar{\mathbf{Z}}^T \quad (6.86)$$

$$E_{\mathbf{y}; \psi} [\tilde{\mathbf{Z}}^T \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)] = \text{diag}(\boldsymbol{\mu}^T) \bar{\mathbf{Z}}^T \quad (6.87)$$

$$E_{\mathbf{y}; \psi} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \tilde{\mathbf{Z}}] = \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \bar{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T) \quad (6.88)$$

$$E_{\mathbf{y}; \psi} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)] = \text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \bar{\mathbf{Z}}^T) \quad (6.89)$$

$$E_{\mathbf{y}; \psi} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)] = \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \bar{\mathbf{Z}}^T \quad (6.90)$$

$$E_{\mathbf{y}; \psi} [\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}}] = \bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \boldsymbol{\mu} \quad (6.91)$$

$$E_{\mathbf{y}; \psi} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}}] = \bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \boldsymbol{\mu} \quad (6.92)$$

$$E_{\mathbf{y}; \psi} [\text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}})] = \text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}} (\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) + \sigma^2 \mathbf{I}_t)),$$

$$\text{where } \mathbf{I}_t \text{ is the identity matrix of order } t. \quad (6.93)$$

$$E_{\mathbf{y}; \psi} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}] = \text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}} (\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) + \sigma^2 \mathbf{I}_t)),$$

$$\text{where } \mathbf{I}_t \text{ is the identity matrix of order } t. \quad (6.94)$$

Proof of Equations (6.73) to (6.76). From the result of Proposition 6.6.1 we have that $E_{\mathbf{y}; \psi} [\tilde{z}_{ijk}]$. Let

$$\bar{\mathbf{Z}} = \begin{bmatrix} \mathbf{1}_{n_1} \mathbf{w}_1^T(\boldsymbol{\phi}) \\ \mathbf{1}_{n_2} \mathbf{w}_2^T(\boldsymbol{\phi}) \\ \vdots \\ \mathbf{1}_{n_s} \mathbf{w}_s^T(\boldsymbol{\phi}) \end{bmatrix}.$$

Then $\bar{\mathbf{Z}}_{(ij)k} = w_{ik}$ and $\bar{\mathbf{Z}} = E_{\mathbf{y}; \psi} [\tilde{\mathbf{Z}}]$. This proves Equation (6.73).

The results given in Equations (6.74) to (6.76) follow directly from Equation (6.73).

□

Proof of Equation (6.77).

$$E_{\mathbf{y}; \psi}[\ddot{\mathbf{y}}] = E_{\mathbf{y}; \psi}[\mathbf{y} - \mathbf{X}_2 \mathbf{b}^*] = E_{\mathbf{y}; \psi}[\mathbf{y}] - \overline{\mathbf{X}}_2 \mathbf{b}^* = \overline{\mathbf{X}}_1 \mathbf{b} = \overline{\mathbf{Z}} \mathbf{C} \mathbf{b} = \overline{\mathbf{Z}} \boldsymbol{\mu}$$

□

Proof of Equations (6.78) and (6.79). The matrix $\tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}}$ is $t \times t$ with the element in row k and column k' equal to

$$[\tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}}]_{kk'} = (\tilde{\mathbf{Z}}_{\bullet k})^T \tilde{\mathbf{Z}}_{\bullet k'} = \sum_{i=1}^s \sum_{j=1}^{n_i} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})$$

Applying the result of Proposition 6.6.2, we obtain,

$$\begin{aligned} E_{\mathbf{y}; \psi}[\tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}}]_{kk'} &= E_{\mathbf{y}; \psi}[(\tilde{\mathbf{Z}}_{\bullet k})^T \tilde{\mathbf{Z}}_{\bullet k'}] \\ &= \begin{cases} 0, & \text{if } k \neq k' \\ \sum_{i=1}^s \sum_{j=1}^{n_i} w_{ik} = \sum_{i=1}^s n_i w_{ik}, & \text{if } k = k', \end{cases} \end{aligned}$$

and so $E_{\mathbf{y}; \psi}[\tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}}] = \text{diag}(\mathbf{1}_n^T \overline{\mathbf{Z}})$.

The result, $E_{\mathbf{y}; \psi}[\tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i] = \text{diag}(\mathbf{1}_{n_i}^T \overline{\mathbf{Z}}_i)$, follows directly. □

Proof of Equation (6.80). If $i \neq i'$, then $\tilde{\mathbf{Z}}_i^T \boldsymbol{\Delta}_i \boldsymbol{\Delta}_{i'}^T \tilde{\mathbf{Z}}_{i'}$ is a $t \times t$ matrix with all elements equal to zero. If $i = i'$, then $\tilde{\mathbf{Z}}_i^T \boldsymbol{\Delta}_i \boldsymbol{\Delta}_{i'}^T \tilde{\mathbf{Z}}_{i'} = \tilde{\mathbf{Z}}_i^T \tilde{\mathbf{Z}}_i$.

Therefore,

$$\begin{aligned} E_{\mathbf{y}; \psi}[\tilde{\mathbf{Z}}_i^T \boldsymbol{\Delta}_i \boldsymbol{\Delta}_{i'}^T \tilde{\mathbf{Z}}_{i'}] &= \begin{cases} 0, & \text{if } i \neq i' \\ \text{diag}(\mathbf{1}_{n_i}^T \overline{\mathbf{Z}}_i), & \text{if } i = i', \end{cases} \\ &= \text{diag}(\mathbf{1}_{n_{i'}}^T \boldsymbol{\Delta}_{i'} \boldsymbol{\Delta}_i^T \overline{\mathbf{Z}}_i). \end{aligned}$$

□

Proof of Equations (6.81) to (6.83). The vector $\mathbf{\ddot{y}}^T \tilde{\mathbf{Z}}$ is a row vector of order t with the k^{th} element equal to

$$(\mathbf{\ddot{y}}^T \tilde{\mathbf{Z}})_{\bullet k} = \mathbf{\ddot{y}}^T \tilde{\mathbf{Z}}_{\bullet k} = \sum_{i=1}^s \sum_{j=1}^{n_i} \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}).$$

We know, from Proposition 6.6.3, that

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi})] = w_{ik} \mu_k.$$

Therefore the k^{th} element of the $1 \times t$ vector $\mathbf{\ddot{y}}^T \tilde{\mathbf{Z}}$ is equal to

$$\sum_{i=1}^s \sum_{j=1}^{n_i} w_{ik} \mu_k = \mu_k \sum_{i=1}^s n_i w_{ik}.$$

The vector $\mathbf{1}_n^T \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T)$ is $1 \times t$ with its k^{th} element equal to $\mu_k \sum_{i=1}^s n_i w_{ik}$. This yields $E_{\mathbf{y}; \boldsymbol{\psi}} [\mathbf{\ddot{y}}^T \tilde{\mathbf{Z}}] = \mathbf{1}_n^T \tilde{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T)$. Therefore, Equation (6.81) is true.

Equation (6.82) follows directly from Equation (6.81) by the definition of the diag function, which is given in Equation (5.53).

By the definition of $\boldsymbol{\Delta}_i$, we have that $\text{diag}(\mathbf{\ddot{y}}^T \boldsymbol{\Delta}_i^T \tilde{\mathbf{Z}}_i) = \mathbf{\ddot{y}}_i^T \tilde{\mathbf{Z}}_i$ and so Equation (6.83) also follows directly from Equation (6.81).

□

Proof of Equations (6.84) and (6.85). The matrix $\tilde{\mathbf{Z}}^T \text{diag}(\mathbf{\ddot{y}}^T) \tilde{\mathbf{Z}}$ is a $t \times t$ matrix with the element in row k and column k' equal to

$$[\tilde{\mathbf{Z}}^T \text{diag}(\mathbf{\ddot{y}}^T) \tilde{\mathbf{Z}}]_{kk'} = (\tilde{\mathbf{Z}}_{\bullet k})^T \text{diag}(\mathbf{\ddot{y}}^T) \tilde{\mathbf{Z}}_{\bullet k'} = \sum_{i=1}^s \sum_{j=1}^{n_i} \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}).$$

We know, from Proposition 6.6.4, that

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})] = \begin{cases} 0 & \text{if } k \neq k' \\ w_{ik} \mu_k & \text{if } k = k'. \end{cases}$$

Therefore,

$$\begin{aligned} E_{\mathbf{y}; \psi}[\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{kk'} &= \begin{cases} 0, & \text{if } k \neq k' \\ \sum_{i=1}^s n_i w_{ik} \mu_k, & \text{if } k = k' \end{cases} \\ &= [\text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T))]_{kk'}. \end{aligned}$$

This proves Equation (6.84).

Equation (6.85) follows directly from Equation (6.84) together with the definition of Δ_i .

□

Proof of Equation (6.86). The matrix $\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T)$ is $t \times n$ with the $k(ij)^{\text{th}}$ element equal to:

$$\begin{aligned} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T)]_{k(ij)} &= [\tilde{\mathbf{Z}}^T]_{k\bullet} [\text{diag}(\ddot{\mathbf{y}}^T)]_{\bullet(ij)} \\ &= \tilde{z}_{ijk} \ddot{y}_{ij} \\ &= \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \end{aligned}$$

Therefore, using Proposition 6.6.3, we have

$$E_{\mathbf{y}; \psi}[\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T)]_{k(ij)} = w_{ik} \mu_k = [\text{diag}(\boldsymbol{\mu}^T) \bar{\mathbf{Z}}^T]_{k(ij)}.$$

□

Proof of Equation (6.87). The matrix $\tilde{\mathbf{Z}}^T \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)$ is $t \times n$ and its $k(ij)^{\text{th}}$ element is equal to:

$$\begin{aligned} [\tilde{\mathbf{Z}}^T \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{k(ij)} &= [\tilde{\mathbf{Z}}^T]_{k\bullet} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{\bullet(ij)} \\ &= \tilde{z}_{ijk} \sum_{k'=1}^t \mu_{k'} \tilde{z}_{ijk'} \\ &= \sum_{k'=1}^t \mu_{k'} \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \end{aligned}$$

From Proposition 6.6.2, we know that

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tau_{ik'}(y_{ij}; \boldsymbol{\psi}) \tau_{ik}(y_{ij}; \boldsymbol{\psi})] = \begin{cases} 0 & \text{if } k \neq k' \\ w_{ik} & \text{if } k = k'. \end{cases}$$

Therefore, $E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}^T \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{k(ij)} = w_{ik} \mu_k = [\text{diag}(\boldsymbol{\mu}^T) \bar{\mathbf{Z}}^T]_{k(ij)}$.

□

Proof of Equation (6.88). The multiplication operation is commutative on diagonal matrices of the same order. Therefore,

$$\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \tilde{\mathbf{Z}} = \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \tilde{\mathbf{Z}}$$

and because $\text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i)$ is a constant matrix,

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \tilde{\mathbf{Z}}] = \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \tilde{\mathbf{Z}}]$$

Taking the transpose of both sides of Equation (6.87) we obtain

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \tilde{\mathbf{Z}}] = \bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T).$$

This proves that $E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \tilde{\mathbf{Z}}] = \text{diag}(\mathbf{1}_{n_i}^T \boldsymbol{\Delta}_i) \bar{\mathbf{Z}}_i \text{diag}(\boldsymbol{\mu}^T)$.

□

Proof of Equation (6.89). The matrix $[\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]$ is $n \times n$ with the element in row (ij) and column $(i'j')$ equal to:

$$\begin{aligned} & [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{(ij)(i'j')} \\ &= [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{(ij)\bullet} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{\bullet(i'j')} \\ &= \begin{cases} 0 & \text{if } (ij) \neq (i'j') \\ \left(\sum_{k=1}^t \mu_k \tilde{z}_{ijk} \right) \left(\sum_{k'=1}^t \mu_{k'} \tilde{z}_{i'j'k'} \right) & \text{if } (ij) = (i'j') \end{cases} \\ &= \begin{cases} 0 & \text{if } (ij) \neq (i'j') \\ \sum_{k=1}^t \sum_{k'=1}^t \mu_k \mu_{k'} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) & \text{if } (ij) = (i'j') \end{cases} \end{aligned} \tag{6.95}$$

Using Proposition 6.6.2, we obtain the following result.

$$\begin{aligned} E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{(ij)(i'j')} &= \begin{cases} 0 & \text{if } (ij) \neq (i'j') \\ \sum_{k=1}^t w_{ik} \mu_k^2 & \text{if } (ij) = (i'j') \end{cases} \\ &= E_{\mathbf{y}; \boldsymbol{\psi}} [\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \bar{\mathbf{Z}}^T)]_{(ij)(i'j')} \end{aligned}$$

□

Proof of Equation (6.90). The matrix $\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)$ is $t \times n$ and its $k(ij)^{\text{th}}$ element is equal to:

$$\begin{aligned} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{k(ij)} &= [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T)]_{k\bullet} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{\bullet(ij)} \\ &= \tilde{z}_{ijk} \ddot{y}_{ij} \sum_{k'=1}^t \mu_{k'} \tilde{z}_{ijk'} \\ &= \sum_{k'=1}^t \mu_{k'} \ddot{y}_{ij} \tilde{z}_{ijk} \tilde{z}_{ijk'} \\ &= \sum_{k'=1}^t \mu_{k'} \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) \end{aligned}$$

Therefore,

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{k(ij)} = \sum_{k'=1}^t \mu_{k'} E_{\mathbf{y}; \boldsymbol{\psi}} [\ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi})]$$

Applying the result of Proposition 6.6.4, we obtain the following.

$$\begin{aligned} E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]_{k(ij)} &= \mu_k^2 w_{ik} \\ &= [\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \bar{\mathbf{Z}}^T]_{k(ij)} \end{aligned}$$

□

Proof of Equation (6.91). The matrix $[\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}}]$ is a column vector of order n with the element in row (ij) given by

$$\begin{aligned} \left[\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}} \right]_{(ij)\bullet} &= \left[\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T) \right]_{(ij)\bullet} \ddot{\mathbf{y}} \\ &= \sum_{k=1}^t \mu_k^2 \tilde{z}_{ijk} \ddot{y}_{ij} \\ &= \sum_{k=1}^t \mu_k^2 \ddot{y}_{ij} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \end{aligned}$$

Therefore, using Proposition 6.6.3, we have

$$\begin{aligned} E_{\mathbf{y}; \boldsymbol{\psi}} \left[\text{diag}(\boldsymbol{\mu}^T \text{diag}(\boldsymbol{\mu}^T) \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}} \right]_{(ij)\bullet} &= \sum_{k=1}^t \mu_k^3 w_{ik} \\ &= \bar{\mathbf{Z}}_{(ij)\bullet} \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \boldsymbol{\mu} \\ &= \left[\bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \boldsymbol{\mu} \right]_{(ij)\bullet} \end{aligned}$$

□

Proof of Equation (6.92). To prove Equation (6.92), we begin with Equation (6.95), which is part of the proof of Equation (6.89).

Equation (6.95) implies that $[\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T)]$ is a $n \times n$ diagonal matrix where the diagonal element in cell (ij, ij) is given by

$$\sum_{k=1}^t \sum_{k'=1}^t \mu_k \mu_{k'} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}).$$

Therefore, the $[\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}}]$ is a column vector of order n with the element in row (ij) given by

$$\left[\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}} \right]_{(ij)\bullet} = \sum_{k=1}^t \sum_{k'=1}^t \mu_k \mu_{k'} \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) \ddot{y}_{ij}.$$

Applying the result of Proposition 6.6.4, we obtain the following.

$$\begin{aligned}
 E_{\mathbf{y}; \psi} [\text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \text{diag}(\boldsymbol{\mu}^T \tilde{\mathbf{Z}}^T) \ddot{\mathbf{y}}]_{(ij)\bullet} &= \sum_{k=1}^t \sum_{k'=k}^t \mu_k^2 \mu_{k'} w_{ik} \\
 &= \sum_{k=1}^t \mu_k^3 w_{ik} \\
 &= \left[\bar{\mathbf{Z}} \text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) \boldsymbol{\mu} \right]_{(ij)\bullet}
 \end{aligned}$$

□

Proof of Equation (6.93). The k^{th} diagonal element of the $t \times t$ diagonal matrix $[\text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}})]$ is, by definition, the k^{th} element of the row vector $\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}$.

$$[\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{\bullet k} = \ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}_{\bullet k} = \sum_{ij} \ddot{y}_{ij} \tau_{ik}(y_{ij}; \psi)$$

Using Proposition 6.6.5, we obtain

$$\begin{aligned}
 E_{\mathbf{y}; \psi} [\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{\bullet k} &= \sum_{ij} w_{ik} (\sigma^2 + \mu_k^2) \\
 &= \sum_{i=1}^s \sum_{j=1}^{n_i} w_{ik} (\sigma^2 + \mu_k^2) \\
 &= \sum_{i=1}^s n_i w_{ik} (\sigma^2 + \mu_k^2) \\
 &= (n_1 w_{i1}, n_2 w_{i2}, \dots, n_t w_{it}) [\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) + \sigma^2 \mathbf{I}_t]_{\bullet k} \\
 &= \mathbf{1}_n^T \bar{\mathbf{Z}} [\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) + \sigma^2 \mathbf{I}_t]_{\bullet k}
 \end{aligned}$$

Therefore,

$$E_{\mathbf{y}; \psi} [\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}] = \mathbf{1}_n^T \bar{\mathbf{Z}} (\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) + \sigma^2 \mathbf{I}_t)$$

and

$$E_{\mathbf{y}; \psi} [\text{diag}(\ddot{\mathbf{y}}^T \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}})] = \text{diag}(\mathbf{1}_n^T \bar{\mathbf{Z}} (\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) + \sigma^2 \mathbf{I}_t)).$$

□

Proof of Equation (6.94). The element in row k and column k' of the $t \times t$ diagonal matrix $\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}$ is given by

$$\begin{aligned} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{kk'} &= (\tilde{\mathbf{Z}}_{\bullet k}^T) \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}_{\bullet k'} \\ &= \sum_{ij} \ddot{y}_{ij}^2 \tau_{ik}(y_{ij}; \boldsymbol{\psi}) \tau_{ik'}(y_{ij}; \boldsymbol{\psi}) \end{aligned}$$

Using Proposition 6.6.6, we obtain

$$\begin{aligned} E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{kk'} &= \begin{cases} 0 & \text{if } k \neq k' \\ \sum_{ij} w_{ik} (\sigma^2 + \mu_k^2) & \text{if } k = k'. \end{cases} \\ &= \begin{cases} 0 & \text{if } k \neq k' \\ \sum_{i=1}^s n_i w_{ik} (\sigma^2 + \mu_k^2) & \text{if } k = k'. \end{cases} \end{aligned}$$

Therefore,

$$E_{\mathbf{y}; \boldsymbol{\psi}} [\tilde{\mathbf{Z}}^T \text{diag}(\ddot{\mathbf{y}}^T) \text{diag}(\ddot{\mathbf{y}}^T) \tilde{\mathbf{Z}}]_{kk'} = \left[\text{diag} \left(\mathbf{1}_n^T \tilde{\mathbf{Z}} (\text{diag}(\boldsymbol{\mu}^T) \text{diag}(\boldsymbol{\mu}^T) + \sigma^2 \mathbf{I}_t) \right) \right]_{kk'}.$$

□

This chapter provided proofs that the information matrix formulae that were presented in Chapter 5 do in fact hold for Normal mixtures. These formulae yield a quick and valid method for estimating the standard errors of the estimated QTL effects and positions generated by CIM and RIM1. In the next chapter, the Fisher information matrix formulae are applied to both the CIM and RIM1 models.

Chapter 7

Simulations and Results

Three new tools were introduced in Chapter 5. These tools are the RIM1 model, the formulae for calculating the Fisher information matrix for Normal mixture models, and a compound hypothesis test for QTL effect and position. Backcross samples were simulated to test the performance of these tools in two different situations. In the first situation, simulations were based on a trait whose value was determined by the genotypes at a single QTL. In the second situation, simulations were based on a trait controlled by many QTL. The same marker-map was used with both cases.

In this chapter, we discuss the details of the simulations and we assess the results. The results demonstrate that our three tools combine to form a robust framework for analysing QTL.

- Our formulae for the information matrix generated good estimates for the standard errors of the MLEs. Moreover, the estimates of standard error became increasingly better with increasing sample size.
- The compound hypothesis test (with standard errors based on the expected information matrix) had the ability to control for the fact that QTL effects and positions test are not separable in the model.

- The compound hypothesis test improved the performance of CIM by dramatically reduced ghosting while retaining strong power to detect QTL.
- The RIM1 procedure performed as well as the improved CIM. The extra QTL fitted in RIM1 made it resistant to ghosting even when a simple test for QTL effect was used instead of the compound hypothesis test.
- For the backcross, samples sizes of 125 yielded unreliable parameter estimates and low power to detect QTL, whereas sample sizes 500 and 2000 yielded estimates of QTL effect and location.
- In the multi-QTL situation the RIM1 procedure was more resistant to ghosting than CIM. However, in the multi-QTL situation both CIM and RIM1 experienced a reduction in power to detect QTL when compared with the power obtained in the single-QTL situation. Similarly, with both methods ghosting was also more likely in the multi-QTL situation than in the single-QTL situation.

7.1 The Single-QTL Situation

An artificial genetic map was defined comprising 35 markers on two chromosomes together with a single QTL. Figure 7.1 is a visual representation of the genetic map that was used in the single-QTL situation. The markers were equally spaced, with a distance of ten centi-Morgans between adjacent markers.

The QTL (labelled *QTL 9*) was placed between markers *c2m7* and *c2m8* on chromosome 2, and its genotypes were used to determine the values of a trait labelled *t1*. The location and effects of *QTL 9* were chosen arbitrarily. The actual location of *QTL 9* was 3.19 centi-Morgans to the right of marker *c2m7*.

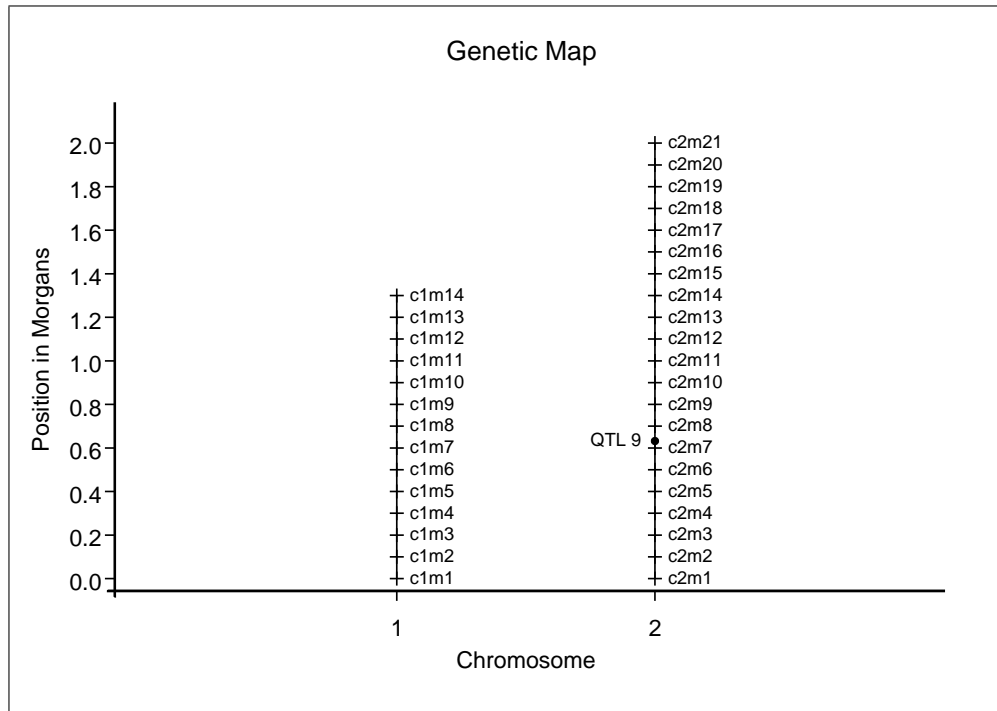


Figure 7.1: Single-QTL, genetic map on which simulations were based.

Haldane's map function was used to convert genetic distances to recombination fractions. Consequently, the recombination fraction between $c2m7$ and $c2m8$ was $r_{MN} = 0.0906$, the recombination fraction between marker $c2m7$ and the QTL was $r_{MQ} = 0.0309$. Therefore, $p_{Q2} = 0.681$ and $p_{Q1} = 0.998$. The additive effects and dominance effects of *QTL 9* were set to $a_{QQ} = 3.14$ and $d_{QQ} = -0.28$ units respectively. Therefore, for the backcross, $b_Q = (a_{QQ} + 2d_{QQ}) = 2.58$ is the only estimable QTL effect and it is associated with the coding scheme $QQ = 1$ and $Qq = 0$.

The QTL Cartographer module Rcross was used to simulate samples from a B1 backcross population. Note that the genetic effects were input into Rcross as the parameters `additive` and `dominance`, with `additive` = $a_{QQ} = 3.14$ and `dominance` = $-2d_{QQ} = 0.56$. Appendix B.3 gives an example of how QTL Cartographer was launched from within the R programming environment.

In Rcross, the broad sense heritability of the trait was taken to be $H^2 = \frac{1}{2}$. The

trait values were determined by taking genotypic values based Cockerham's (1954) linear model and adding a random variable having mean zero and variance equal to σ^2 , where σ^2 is determined by the heritability ($H^2 = \frac{1}{2}$) and the genotypic variance.

By definition,

$$H^2 = \frac{\text{var}(G)}{\sigma^2 + \text{var}(G)}$$

where $\text{var}(G)$ is the variance of the genotypic values. The genotype probabilities ($P(QQ) = \frac{1}{2}$, $P(Qq) = \frac{1}{2}$) in backcross and the genotypic effects ($a_{QQ} = 3.14$, $d_{QQ} = -0.28$) imply that the expected value of the genotypic variance in this backcross population is

$$\begin{aligned} \text{var}(G) &= \frac{1}{2}(\mu_{QQ}^2 + \mu_{Qq}^2) - \frac{1}{4}(\mu_{QQ} + \mu_{Qq})^2 \\ &= \frac{1}{4}a_{QQ}^2 + a_{QQ}d_{QQ} + d_{QQ}^2 = 1.664 \text{ to three decimal places.} \end{aligned}$$

Therefore, because $H^2 = \frac{1}{2}$, the expected value of σ^2 in the simulated data is

$$\sigma^2 = \text{var}(G) = 1.664.$$

The following samples were all simulated from the same B1 backcross population.

- One hundred replicate samples each containing 125 individuals.
- One hundred replicate samples each containing 500 individuals.
- One hundred replicate samples each containing 2000 individuals.

For a fixed sample size, analysis of replicate samples from the same distribution represents a Monte-Carlo experiment from which we can estimate the distribution of any model parameter estimator.

All samples were imported into R objects to facilitate analysis with RIM1 and calculation of the information matrix for CIM and RIM1. The aim of the analysis was to scan chromosome 2 to test for the existence of a QTL and to estimate QTL

location and effect. As this is a simulation study, the true properties of the QTL are known, so we can assess model performance by comparing model parameters to their true values. Table 7.1 gives an example of raw output from our implementation of RIM1 for a sample of size 2000.

Table 7.1: An Example of raw output from our RIM1 implementation

```
> b1c2.LQR.neut7[[1]] #display RIM1 output for the first sample.
$code
[1] "rim.linecross()"

$information.matrix
[1] "expected"

$chosen.model.desc
[1] "RIM1"

$chosen.model
[1] "LQR"

$mapfun
[1] "Haldane"

$cross
[1] "B1"

$interval
[1] "c2m7" "c2m8"

$markers
[1] "c2m6" "c2m7" "c2m8" "c2m9"

$extra.markers
[1] "c1m1" "c1m2" "c1m3" "c1m4" "c1m5" "c1m6" "c1m7"
[8] "c1m8" "c1m9" "c1m10" "c1m11" "c1m12" "c1m13" "c1m14"
[15] "c2m1" "c2m2" "c2m3" "c2m4" "c2m5" "c2m10" "c2m11"
[22] "c2m12" "c2m13" "c2m14" "c2m15" "c2m16" "c2m17" "c2m18"
[29] "c2m19" "c2m20" "c2m21"

$trait
[1] "t1"
```

Table 7.1: An Example of raw output from RIM1 (continued)

```

$genotype.counts
AAAAAAAA AAAAAAaA AAAAAaAA AAAAAaAa AAAaAAAA AAaAAAAa
      721      87      13      82      4      1
AAAAaAaAA AAAaAaAa AaAAAAAA AaAAAAAa AaAAAAaA AAaAAAAaA
      13      62      86      17      1      6
AaAaAAAAA AaAaAAAAA AaAaAaAA AaAaAaAa
      68      10      95      734

$map.hat
      rKM      rMN      rNO
0.0950 0.0925 0.1185

$mle
$mle$convergence.info
chosen.tolerance actual.tolerance  num.iterations
      1.00e-06      6.88e-07      2.50e+01

$mle$model.params
$mle$model.params$effects
              MLE std.err      z0 P>|z0|
(Intercept)  0.1637  0.0750  2.183 0.0290
L.AA          0.0415  0.1192  0.348 0.7277
Q.AA          2.7397  0.1020 26.866 0.0000
R.AA         -0.1100  0.1078 -1.020 0.3079
c1m1.AA      -0.0315  0.0977 -0.322 0.7472
c1m2.AA       0.1951  0.1272  1.534 0.1250
.
.
.
c2m21.AA     -0.0073  0.1014 -0.072 0.9426

$mle$model.params$variance
MLE
1.68

$mle$model.params$probs
              MLE std.err      z0      z1 P>z0      P<z1
pL2 0.99999 0.000246 4.07e+03 -5.11e-02 0.00 4.80e-01
pQ2 0.66548 0.034504 1.93e+01 -9.70e+00 0.00 1.58e-22
pR2 0.00001 0.000198 5.04e-02 -5.04e+03 0.48 0.00e+00

$mle$infmtat.is.singular
[1] FALSE

```

Table 7.1: An Example of raw output from RIM1 (continued)

<code>\$mle\$recomb</code>					
<code>rMQ</code>	<code>rQN</code>	<code>rKL</code>	<code>rLM</code>	<code>rKM</code>	<code>rNR</code>
3.23e-02	6.23e-02	1.25e-06	9.06e-02	9.06e-02	9.06e-02
<code>rR0</code>	<code>rN0</code>	<code>rMN</code>	<code>pQ1</code>	<code>pL1</code>	<code>pR1</code>
9.97e-07	9.06e-02	9.06e-02	9.98e-01	1.00e+00	1.00e+00
 <code>\$mle\$loglike</code>					
[1] -3417					
<code>\$mle\$startlike</code>					
[1] -3447					
.					
.					
.					

All extra markers were included as cofactors in the RIM1 and CIM models. The samples were also analysed using the QTL Cartographer module ZmapQTL. Program code for importing QTL Cartographer Rcross and ZmapQTL output files into R objects is provided in Appendix B.2. Output from CIM-QTLcart (ZmapQTL running CIM) and from QTL Cartographer's implementations of Lander and Botstein's interval mapping (IM) were compared with output from the RIM1 model and with output from our implementation of CIM. The output was summarised as shown in Tables 7.2 and 7.3 for four samples.

The b_Q values are given as `Q.AA` in Table 7.2 and as `H1.a` in Table 7.3 and these values are similar (ranging from 2.549 to 2.74 - all close to the true effect 2.58). Also, RIM1 returned σ^2 ranging from 1.631 to 1.678 (all close to the the expected variance of 1.664). RIM1 returned QTL locations \hat{r}_{MQ} ranging from 0.031 to 0.039. To interpret the map distances returned by ZmapQTL in terms of recombination fractions, we use the inverse of Haldane's map function to obtain: $\hat{r}_{MQ} = \frac{1}{2}(1 - \exp^{-2(\text{position}-0.6)})$, where 0.6 is the position of *c2m7* from the left telomere. Therefore, for sample size 2000, both RIM1 and CIM-QTLcart produced values, \hat{r}_{MQ} , that were close to the true r_{MQ} of 0.0309.

Table 7.2: Summarising the output of RIM1 (an example). The estimates shown are $L.AA = \hat{b}_L$, $Q.AA = \hat{b}_Q$, $R.AA = \hat{b}_R$, $pL2 = \hat{p}_{L2}$, $pQ2 = \hat{p}_{Q2}$, $pR2 = \hat{p}_{R2}$, $\text{sigma2} = \hat{\sigma}^2$, and $rMQ = \hat{r}_{MQ}$. The logarithm of the likelihood function is loglike . The estimated asymptotic standard deviations of \hat{b}_Q and \hat{p}_{Q2} , are sd.bQ and sd.pQ2 respectively.

```
>#RIM1 output was collected into a list of matrices (named RIM1.nw.all).
>#The sample size is 2000 and the testing interval is c2m7-c2m8.
>#Here is a summary of RIM1 output for samples 1, 2, 3 and 100.
>
> round(RIM1.nw.all[[7]][c(1:3,100),c(5:7,39:43,55:57)],3)
      L.AA  Q.AA  R.AA pL2  pQ2 pR2 sigma2  rMQ  loglike  sd.bQ sd.pQ2
[1,] 0.042 2.740 -0.110  1 0.665  0  1.678 0.032 -3417.161  0.102  0.035
[2,] 0.101 2.549 -0.269  1 0.675  0  1.618 0.031 -3386.591  0.104  0.033
[3,] 0.048 2.672 -0.068  1 0.600  0  1.637 0.038 -3400.221  0.106  0.036
[4,] 0.031 2.636 -0.049  1 0.593  0  1.631 0.039 -3403.081  0.106  0.036
>
>#Use all 100 RIM replicates to estimate the standard deviation (SD) of bQ.
> sqrt(var(RIM1.nw.all[[7]][, "Q.AA"]))
[1] 0.1116254
>
>#Inspect the 100 values for SD of bQ obtained from the information matrix.
> summary(RIM1.nw.all[[7]][, "sd.bQ"])
      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
 0.0978  0.1020  0.1044  0.1044  0.1066  0.1105
```

Table 7.3: Summarising the QTL Cartographer output for CIM (an example). Chromosome = c; m = left marker; position = distance in Morgans of Q from the left telomere; $H0.H1$ is the LRT statistic for $\frac{H_0}{H_1}$; $R2.0.1 = \frac{\hat{\sigma}_0^2 - \hat{\sigma}^2}{\text{var}(y)}$, where $\hat{\sigma}_0^2$ is the residual variance under H_0 ; $TR2.0.1 = \frac{\text{var}(y) - \hat{\sigma}^2}{\text{var}(y)}$; $H1.a = \hat{b}_Q$; $S1 = \frac{nk_3^2}{6} + \frac{nk_4^2}{24}$ where $k_3 = \frac{n^2 E(\varepsilon - \bar{\varepsilon})^3}{(n-1)(n-2)S^3}$, $k_4 = \frac{n^2(n+1) E(\varepsilon - \bar{\varepsilon})^4}{(n-1)(n-2)(n-3)S^4} - 3$, with $\varepsilon = y - \hat{y}$, $\bar{\varepsilon} = E(\varepsilon)$, $S^2 = \frac{n}{n-1} \hat{\sigma}^2$, $n = 2000$.

```
>#Zmapqtl output was imported an R object named 'CIM.all'.
>#CIM.all is a list of matrices (one matrix for each testing interval).
>#Each matrix stores the MLE position obtained by Zmapqtl for each sample.
>#The sample size is 2000 and the testing interval is c2m7-c2m8.
>#Here is a summary of Zmapqtl output for samples 1, 2, 3 and 100
>
> round(CIM.all[[7]][c(1:3,100),],3)
      sample c m position  H0.H1  R2.0.1  TR2.0.1  H1.a  S1
[1,]      1 2 7      0.63 489.891  0.173    0.516 2.742 4.366
[2,]      2 2 7      0.63 391.869  0.147    0.510 2.549 1.302
[3,]      3 2 7      0.64 402.720  0.151    0.513 2.672 0.026
[4,]     100 2 7      0.64 386.756  0.148    0.519 2.636 0.194
>
>#Use all 100 CIM QTLcart replicates to estimate the SD of bQ.
> sqrt(var(CIM.all[[7]][, "H1.a"]))
[1] 0.1120043
>
>#The information matrix is not available from QTL Cartographer.
```

As there are no non-genetic factors, the variable TR2.0.1 from ZmapQTL is an estimate of the broad sense heritability. The values for TR2.0.1 in Table 7.3 range from 0.510 to 0.519, so they closely estimate H^2 which is equal to a half. The variable S1 from ZmapQTL is a statistic based on the coefficients of skewness and kurtosis, and it is used to test for normality of the residuals (Basten *et al.*, 2001, pages 48-49).

The next few sections examine the performance of RIM1 and CIM in greater detail. This includes a discussion of the impact of sample size on the behaviour of the information matrix, on the quality of the maximum likelihood estimates, and on the performance of our hypothesis tests. The replicate samples allowed calculation of empirical estimates for the standard errors of model parameters and calculation of power and rates of false detections associated with hypothesis tests. Section 7.1.2 assesses the behaviour of the estimated Fisher information matrix by comparing standard errors generated by its inverse with empirical standard errors. Section 7.1.1 discusses the quality of the MLEs of QTL effect and position, while Section 7.1.3 assesses the power of our hypothesis test to detect QTL and its robustness against false detections.

7.1.1 Quality of the MLEs of QTL effect and position

The RIM1 model is a new extension and generalisation of Composite Interval Mapping (CIM). Therefore, it is necessary to test whether this extension constitutes an improvement over CIM. To ensure that the comparisons were as objective as possible, the popular QTL Cartographer software was used to simulate all samples and to analyse those samples via composite interval mapping.

Estimation of the standard errors of the MLEs generated by QTL Cartographer's implementation of CIM is not possible without resorting to re-sampling techniques. In this discussion we will refer to QTL Cartographer's implementation of CIM as CIM-QTLcart. The data was also analysed using my own implementation of Zeng's CIM model (referred to as CIM in this discussion).

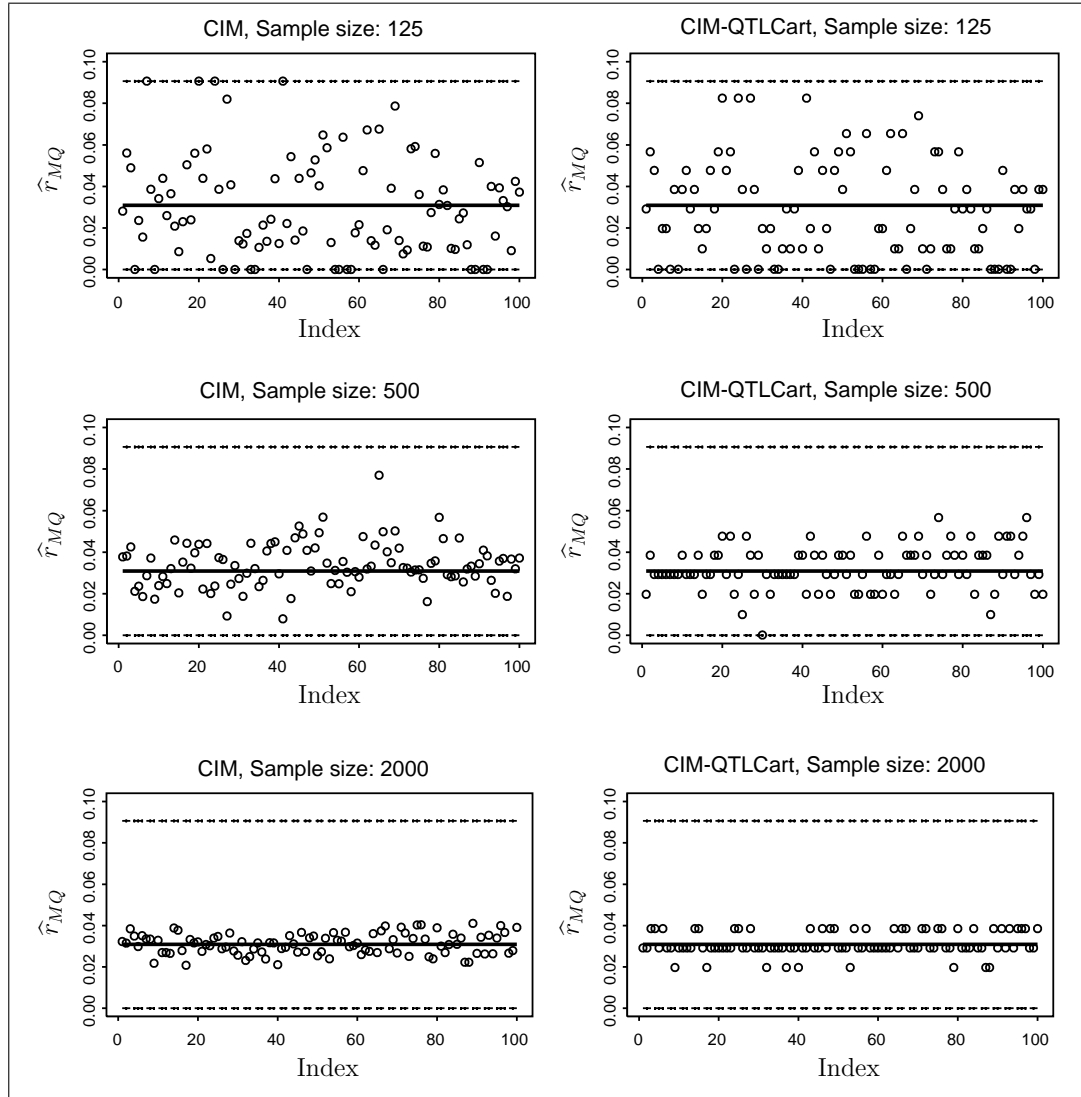


Figure 7.2: Scatter plots of \hat{r}_{MQ} from simulated data ($M = c2m7$). The solid horizontal line at 0.0309 is the true r_{MQ} . Dashed horizontal lines at $\hat{r}_{MQ} = 0$ and $\hat{r}_{MQ} = r_{MN} = 0.0906$ are bounds on r_{MQ} . The estimates \hat{r}_{MQ} were generated by R program code from this thesis (CIM) and by QTL Cartographer (CIM-QTLcart). QTL Cartographer selects the MLE from a discrete grid, leading to points aligned in rows in the graphs on the right.

The motivation for programming an extended implementation of CIM was to have a procedure which calculates the information matrix and which allows the EM algorithm to follow its own native trajectory when moving from a starting point. By comparison, CIM-QTLcart restricts maximization to a fixed grid.

For interval $c2m7 - c2m8$, Figure 7.2 shows the maximum likelihood estimates of the recombination fraction r_{MQ} produced by both implementations of CIM. In the scatter plots, each index value identifies a sample and the corresponding value of \hat{r}_{MQ} is the MLE generated from that sample.

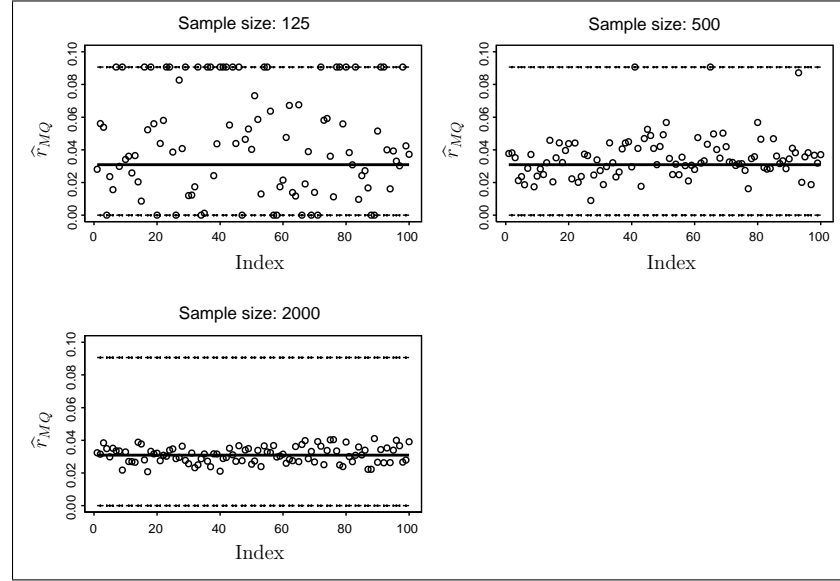
QTL Cartographer selects the MLE from a discrete grid, leading to points aligned in rows in the graphs on the right of Figure 7.2. The difference between the two implementations is only noticeable when a comparison is made between the structured appearance of the points generated by QTL Cartographer and the random appearance of the points generated by our extended implementation of CIM. Still, it is clear from Figure 7.2 that the scatter of points is roughly the same for both implementations of CIM at each sample size.

The most interesting pattern seen in Figure 7.2 is the effect of sample size on the ability of CIM to locate QTL. As the sample size increased from 125 to 2000, the MLEs of r_{MQ} became increasingly stable, settling towards its true value.

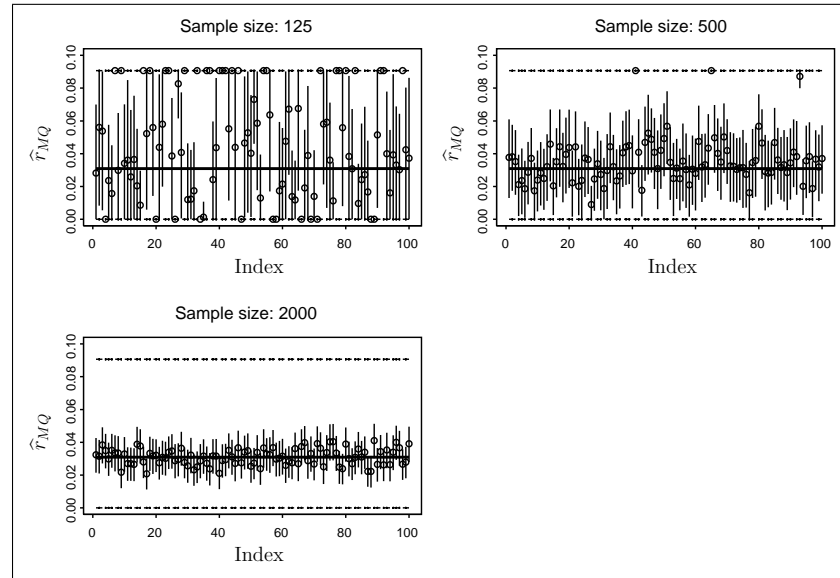
Figure 7.3(a) shows that the MLEs of QTL location generated by applying RIM1 to the same data. In interval $c2m7 - c2m8$, the behaviour of RIM1 was almost identical to that of CIM with the precision of the estimated QTL location improving with increasing sample size. The MLEs of r_{MQ} were calculated from the MLEs of p_{Q2} using the relationships given in equations (5.16) and (5.22). The function

$$\text{recomb}(p_{Q2}) = r_{MN}(1 - p_{Q2}) + 0.5(1 - r_{MN}) - 0.5\sqrt{1 - 2r_{MN} + r_{MN}^2(1 - 2p_{Q2})^2}$$

calculates r_{MQ} given r_{MN} and p_{Q2} . If (a, b) is a $(1 - \alpha)100\%$ confidence interval (CI) for \hat{p}_{Q2} , then the corresponding $(1 - \alpha)100\%$ CI for \hat{r}_{MQ} is $(\text{recomb}(b), \text{recomb}(a))$.



(a) RIM1: Points \hat{r}_{MQ} from 100 replicates at interval c2m7-c2m8.



(b) RIM1: Points \hat{r}_{MQ} from 100 replicates at interval c2m7-c2m8, each with a vertical line stretching over a 99.9% confidence interval (CI) for that point. The CI were derived using the Fisher Information Matrix formulae given in this thesis.

Figure 7.3: Scatter plots of \hat{r}_{MQ} based on simulated data having only one QTL between $M = c2m7$ and $N = c2m8$. The solid horizontal line at 0.0309 is the true r_{MQ} . Dashed horizontal lines at $\hat{r}_{MQ} = 0$ and $\hat{r}_{MQ} = r_{MN} = 0.0906$ are bounds on r_{MQ} . The estimates \hat{r}_{MQ} were generated by the RIM1 model.

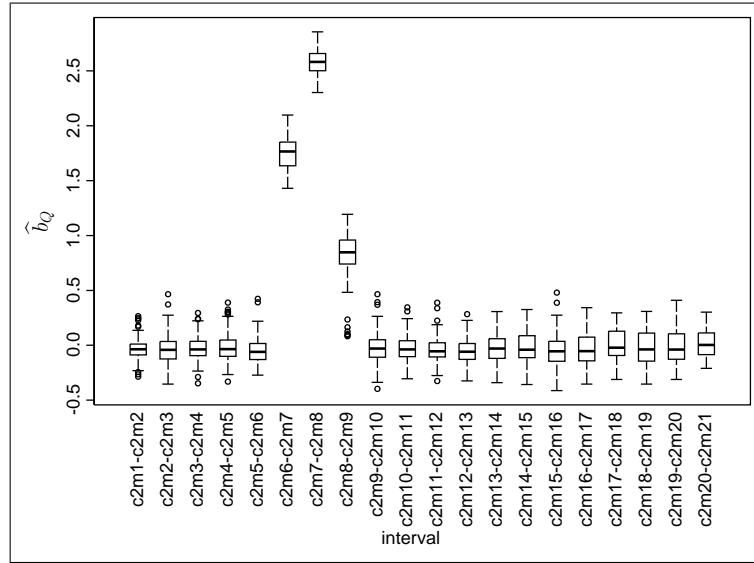
Confidence intervals for \hat{p}_{Q2} were calculated by assuming Normality of the maximum likelihood estimator. Figure 7.3(b) is a scatter plot of \hat{r}_{MQ} with confidence intervals superimposed unto each point. These plots show that as sample size increases, not only do the estimates of QTL location become closer to the true location, but the confidence intervals become shorter as well.

At sample size 125, the confidence intervals generally stretched over the entire length of the marker interval. The width of most confidence intervals for r_{MQ} was around 0.04 recombination units in samples of size 500, and around 0.02 recombination units in samples of size 2000. By Haldane's map function, this corresponds to distances of roughly 4 centi-Morgans wide (sample size 500), and 2 centi-Morgans wide (sample size 2000).

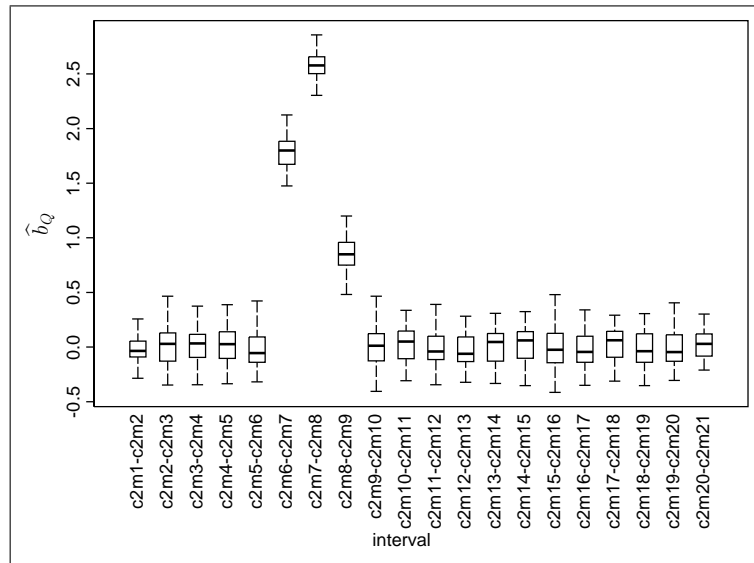
The results show that RIM1 performs as well as CIM in terms of estimating QTL location. However, an estimated location is only meaningful if the data supports the existence of a QTL associated with the markers bordering a testing interval. Detection of significant QTL effect is needed before the question of location can be considered. The strength of RIM1 over CIM is revealed by looking at the estimated QTL effects.

The box plots in Figure 7.4 display the distributions of estimated QTL effects generated by CIM and CIM-QTLcart for samples of size two thousand. Estimates of b_Q were plotted for each interval on the second chromosome. Both implementations of Composite Interval Mapping produced three peaks: one peak in interval $c2m6 - c2m7$, one peak in interval $c2m7 - c2m8$ and one peak in interval $c2m8 - c2m9$.

We know that there is only one QTL on the map. It is named *QTL 9* and is located within interval $c2m7 - c2m8$. The range of values for b_Q within the central peak indicate that CIM generated estimates \hat{b}_Q that were close to $b_Q = 2.58$, the true effect of *QTL 9*.

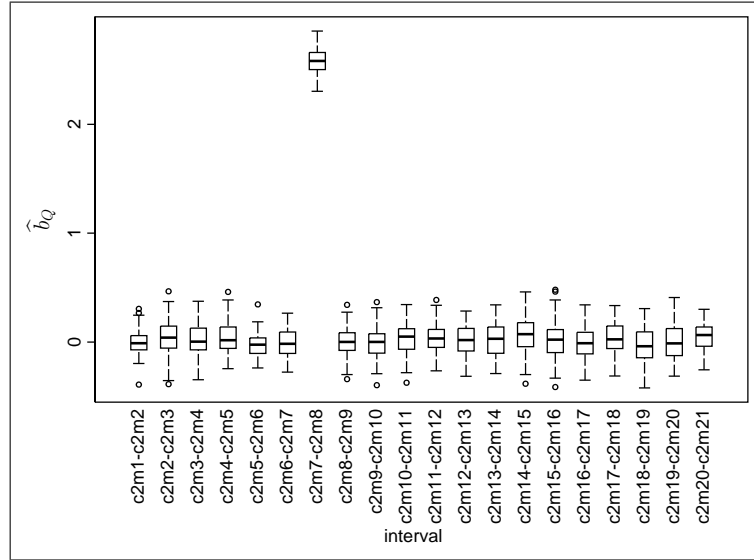


(a) CIM: Sample size 2000, Boxplots of \hat{b}_Q based on 100 replicates at each interval.

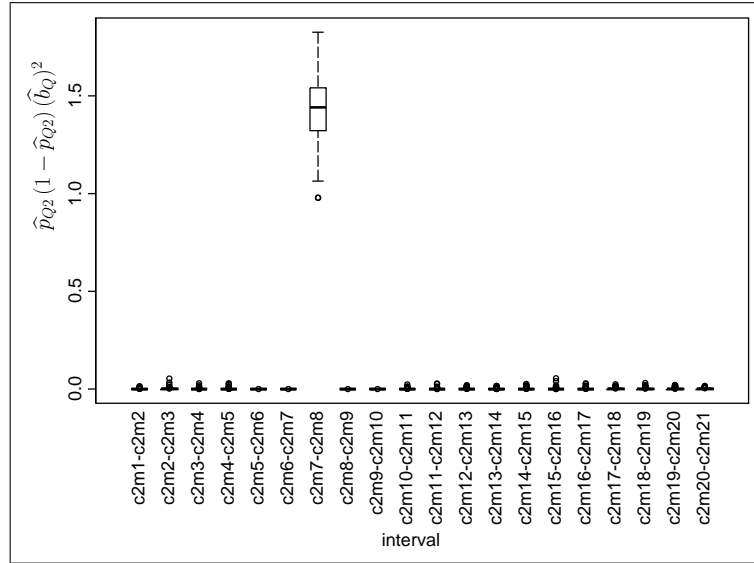


(b) CIM-QTLcart: Sample size 2000, Boxplots of \hat{b}_Q based on 100 replicates at each interval.

Figure 7.4: Box plots of \hat{b}_Q and from CIM and CIM-QTLcart, based on simulated samples with a single QTL and sample size 2000. Each plot shows the upper quartile, median and lower quartile. Whiskers are drawn to the nearest value not beyond $1.5 \times (\text{Inter-Quartile Range})$ from the quartiles; points beyond (outliers) are plotted individually.



(a) RIM1: Sample size 2000, Boxplots of \hat{b}_Q based on 100 replicates at each interval.



(b) RIM1: Sample size 2000, Boxplots of $\hat{p}_{Q2}(1 - \hat{p}_{Q2})(\hat{b}_Q)^2$ based on 100 replicates at each interval.

Figure 7.5: Box plots of \hat{b}_Q and $\hat{p}_{Q2}(1 - \hat{p}_{Q2})(\hat{b}_Q)^2$ from RIM1 based on simulated samples with a single QTL and sample size 2000.

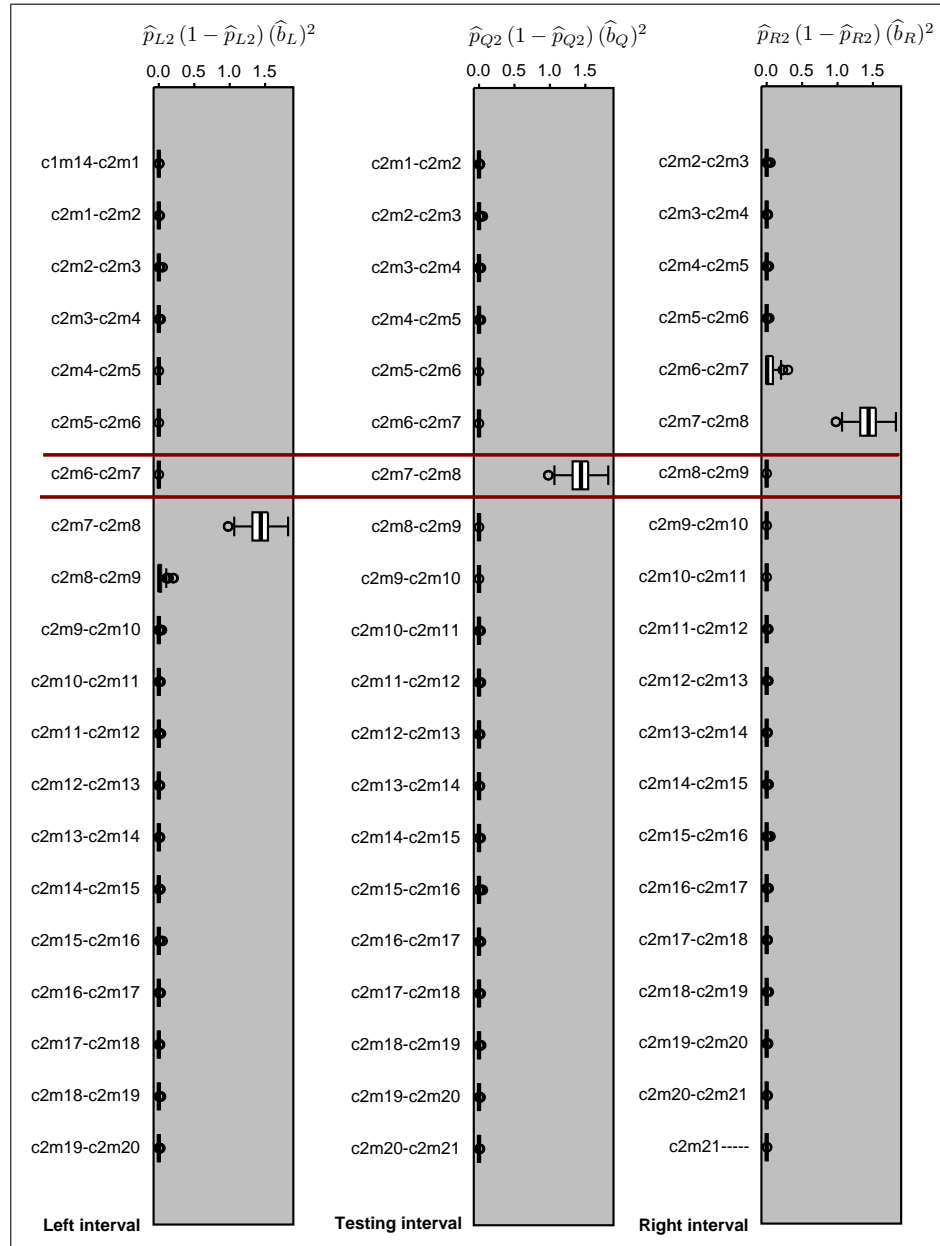


Figure 7.6: Box plots of the estimates $\hat{p}_{L2}(1 - \hat{p}_{L2})(\hat{b}_L)^2$, $\hat{p}_{Q2}(1 - \hat{p}_{Q2})(\hat{b}_Q)^2$ and $\hat{p}_{R2}(1 - \hat{p}_{R2})(\hat{b}_R)^2$ obtained by applying RIM1 to 20 consecutive intervals spanning chromosome 2. RIM1 simultaneously fits three QTL (L , Q , R). Each box plot is based on estimates from 100 replicate samples each of size 2000. There is a single QTL, interior to the interval $c2m7 - c2m8$. These plots show that, for large sample sizes, RIM1 can correctly locate QTL within the left or right adjacent intervals as well as QTL within the central testing interval.

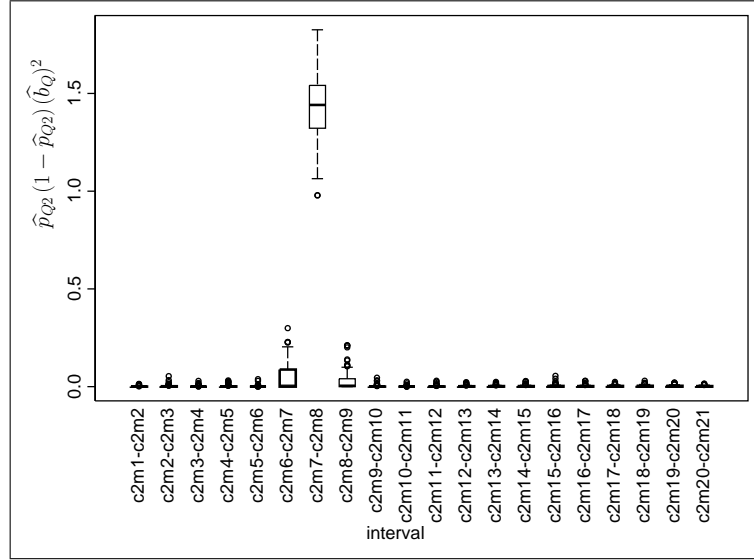
The two extra peaks generated by Composite Interval Mapping are due to the confounding of effects in the testing interval with effects from QTL in an adjacent interval. These two false peaks illustrate the well known ghosting behaviour of CIM.

The box plots in Figure 7.5(a) display the distributions of estimated QTL effects generated by RIM1 for samples of size two thousand. Like CIM, RIM1 generated a peak in the interval $c2m7 - c2m8$. Also like CIM, RIM1 generated estimates for b_Q that were close to the true effect of *QTL 9* with some slight overestimation.

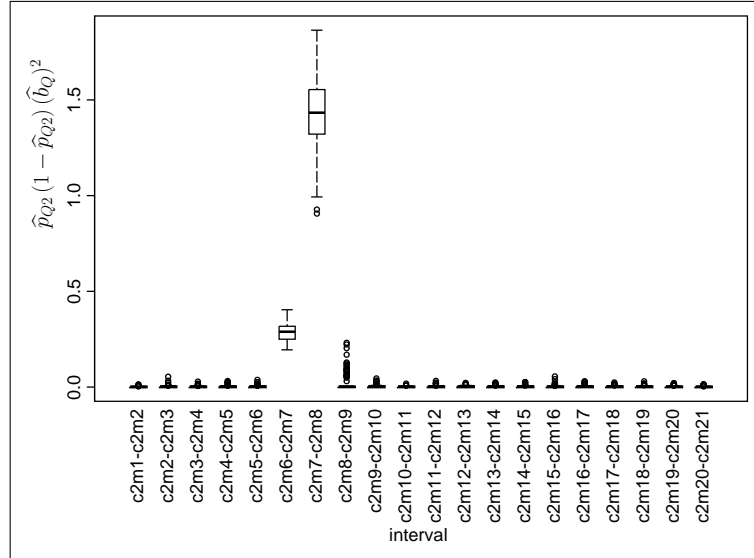
In contrast to the three peaks generated by CIM, only one peak was generated by RIM1. Moreover, this peak was in the interval containing QTL and the estimates of b_Q for all other intervals were close to zero. Figure 7.5(a) demonstrates that RIM1 is better able to separate QTL effects in nearby intervals than CIM. By fitting nuisance parameters (b_L, b_R, p_{L2}, b_{L2}) associated with QTL to the right and left of the testing interval, RIM1 gains a dramatic reduction in ghosting while retaining power to detect QTL.

An idea proposed in Section 5.3 was to use a joint test for QTL effect and position as a strategy to reduce ghosting both in CIM and RIM1. The plots of the quantity $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ displayed in figures 7.5(b), 7.7(a) and 7.7(b) help to explore this idea. If ($b_Q = 0$) or ($p_{Q2} = 0$) or ($p_{Q2} = 0$) then $p_{Q2} (1 - p_{Q2}) (b_Q)^2$ is equal to zero. The plots of $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ in Figures 7.7(a) and 7.7(b) illustrate a dampening the false peaks generated by CIM. By jointly examining QTL effect and location, we can improve the distinction between intervals that contain QTL and those that do not (see Figure 7.6).

Results of RIM1 for samples sizes 500 and 125 are given in Figures 7.8 and 7.9 respectively. We see that sample size impacts on the quality of the estimates of QTL effect in much the same way that it impacts on the quality of the estimates of QTL location. Very small sample sizes (such as sample size 125) can produce spurious results, leading to increased ghosting and loss of power to detect QTL.

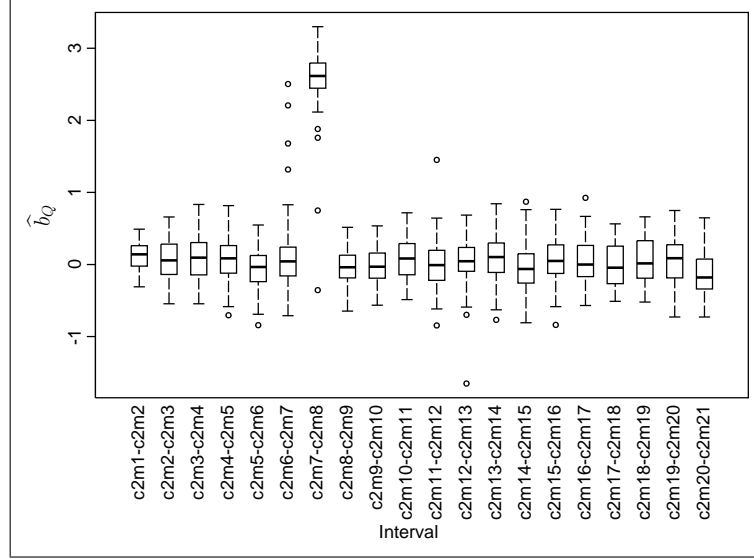


(a) CIM: Sample size 2000, Boxplots of $\hat{p}_{Q2}(1-\hat{p}_{Q2})(\hat{b}_Q)^2$ based on 100 replicates at each interval.

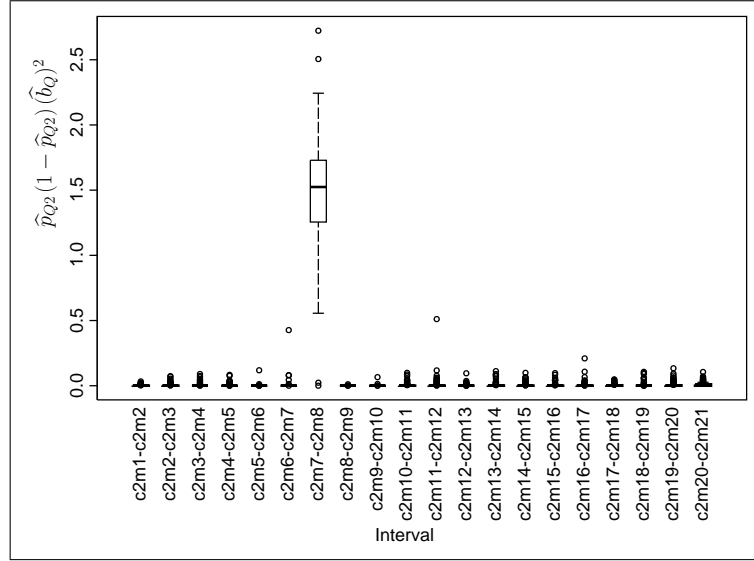


(b) CIM-QTLcart: Sample size 2000, Boxplots of $\hat{p}_{Q2}(1-\hat{p}_{Q2})(\hat{b}_Q)^2$ based on 100 replicates at each interval.

Figure 7.7: Box plots of $\hat{p}_{Q2}(1-\hat{p}_{Q2})(\hat{b}_Q)^2$ from CIM and CIM-QTLcart, based on simulated samples with a single QTL and sample size 2000.

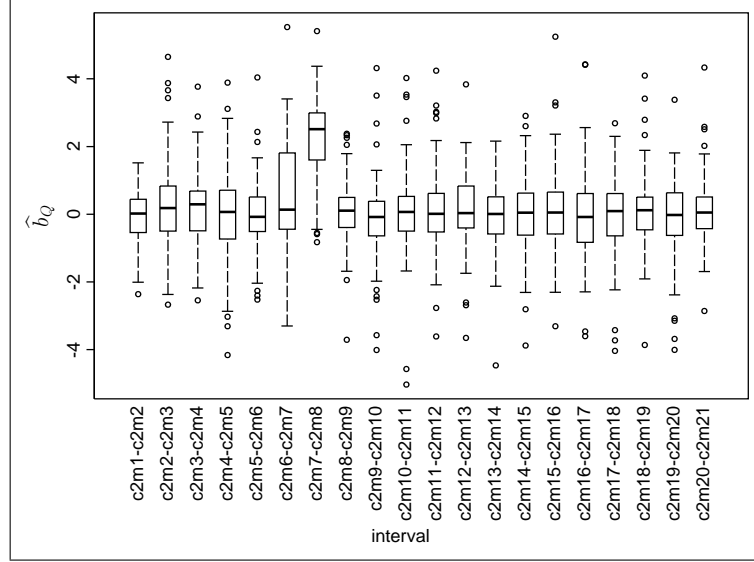


(a) RIM1: Sample size 500, Boxplots of \hat{b}_Q based on 100 replicates at each interval.

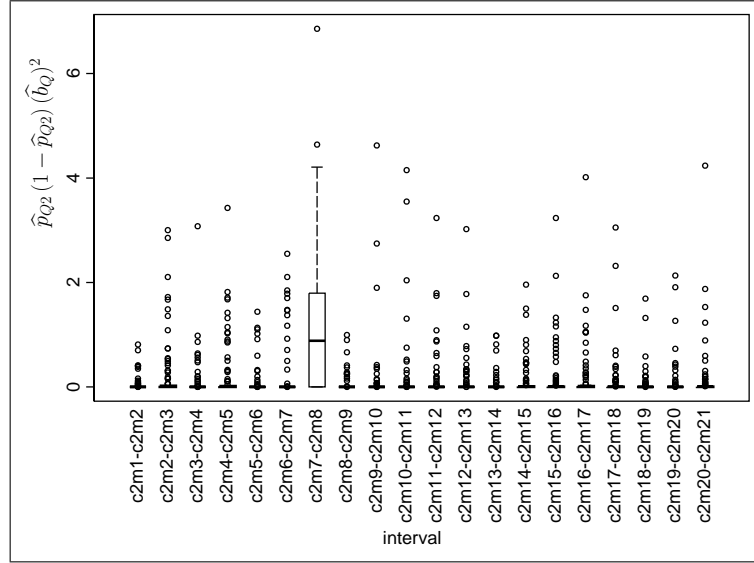


(b) RIM1: Sample size 500, Boxplots of $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ based on 100 replicates at each interval.

Figure 7.8: Box plots of \hat{b}_Q and $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ from RIM1 based on simulated samples with a single QTL and sample size 500.



(a) RIM1: Sample size 125, Boxplots of \hat{b}_Q based on 100 replicates at each interval.



(b) RIM1: Sample size 125, Boxplots of $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ based on 100 replicates at each interval.

Figure 7.9: Box plots of \hat{b}_Q and $\hat{p}_{Q2} (1 - \hat{p}_{Q2}) (\hat{b}_Q)^2$ from RIM1 based on simulated samples with a single QTL and sample size 125.

7.1.2 Performance of the Fisher information matrix

The formula for calculating the information matrix as given in Equation (5.97) is a new development. Moreover, its derivation is not dependent on any specific mixture of normals. Therefore, we have developed a flexible tool for calculating standard errors of maximum likelihood estimates in RIM1, CIM and in any mixture of univariate normals.

We now validate the performance of different methods of standard error estimation using the 100 replicate samples which we have for each of the three sample sizes (125, 500 and 2000). Estimates calculated from these replicates allow us to form an empirical approximation of the sampling distribution of those estimates, which we can use to make these comparisons.

Comparisons with replicates

Take N samples of equal size from the same population. Then Equation (7.1) gives an empirical estimator for the standard error (standard deviation or SD) of \hat{b}_Q based upon the replicate samples. In Equation (7.1) the expression $\hat{b}_Q^{(i)}$ denotes the MLE of b_Q obtained from the i^{th} replicate sample.

$$\text{emp SD of } \hat{b}_Q = \sqrt{\frac{1}{N-1} \left(\sum_{i=1}^N \left(\hat{b}_Q^{(i)} \right)^2 - \frac{1}{N} \left(\sum_{i=1}^N \hat{b}_Q^{(i)} \right)^2 \right)} \quad (7.1)$$

If we have a sample of sufficiently large size, then asymptotic likelihood theory provides an estimator for the standard error of \hat{b}_Q via the inverse of the Fisher information matrix. We estimate the expected information matrix $\mathcal{I}(\psi)$ by using Equation (5.97) and substituting maximum likelihood estimates of the model parameters for their true values. Then Equation (7.2) gives an estimator for the asymptotic standard error of \hat{b}_Q based upon a single sample.

$$\text{imat SD of } \hat{b}_Q = \sqrt{[\mathcal{I}(\hat{\psi})]_{b_Q b_Q}^{-1}} \simeq \sqrt{\text{asymptotic var}(\hat{b}_Q)} \quad (7.2)$$

For 125, 500 and 2000, respectively, and with interval $c2m7 - c2m8$ as the testing interval, Table 7.4 compares the standard errors of \hat{b}_Q from the expected information matrix with the corresponding empirical standard errors.

Table 7.4: Simulated Single-QTL Case: Interval $c2m7$ - $c2m8$; comparison of estimated standard errors (SD) of \hat{b}_Q based on one hundred replicates at each sample size.

Sample size	Model	emp SD of \hat{b}_Q	imat SD of \hat{b}_Q					
125	RIM1	1.350	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	CIM	0.547	0.273	0.391	0.425	0.435	0.463	0.803
	CIM-QTLcart.	0.550	0.273	0.362	0.411	0.414	0.447	0.687
			not applicable					
500	RIM1	0.443	0.191	0.205	0.211	0.212	0.215	0.305
	CIM	0.261	0.191	0.205	0.211	0.210	0.215	0.225
	CIM-QTLcart.	0.222	not applicable					
2000	RIM1	0.112	0.098	0.102	0.104	0.104	0.107	0.111
	CIM	0.112	0.098	0.102	0.104	0.104	0.107	0.111
	CIM-QTLcart.	0.112	not applicable					

At sample sizes 125, 500 and 2000, the empirical standard errors of QTL effect from CIM and CIM-QTLcart were within one decimal place of each other. The information matrix formula gave very stable results with CIM. At sample size 125 CIM had good agreement between the information matrix estimates for the SD of \hat{b}_Q and the empirical estimates of SD. At sample sizes 500 and 2000, CIM had almost perfect agreement amongst the empirical standard errors and those from the information matrix.

At sample size 2000, the models CIM, CIM-QTLcart and RIM1 all gave the same value (0.112) for the empirical SD of the MLE of QTL effect. RIM1 also showed reasonable agreement at sample size 500. However the information matrix severely underestimated the standard errors when the RIM1 model was used with a sample of size 125. RIM1 seemed more sensitive to small sample sizes than CIM. This is not surprising because in the backcross, the RIM1 models has to estimate four more parameters (p_{L2}, p_{R2}, b_L, b_R) than CIM.

Table 7.5: MLE \hat{b}_Q and its estimated standard error from RIM1 on replicates

Interval	Sample size = 125			Sample size = 500			Sample size = 2000		
	mean MLE \hat{b}_Q	mean imat SD	emp SD	mean MLE \hat{b}_Q	mean imat SD	emp SD	mean MLE \hat{b}_Q	mean imat SD	emp SD
c2m1 - c2m2	-0.09	0.38	0.74	0.12	0.19	0.19	0.00	0.10	0.11
c2m2 - c2m3	0.22	0.48	1.25	0.06	0.24	0.28	0.04	0.12	0.15
c2m3 - c2m4	0.16	0.49	1.06	0.09	0.25	0.29	0.02	0.12	0.14
c2m4 - c2m5	0.05	0.49	1.39	0.07	0.25	0.27	0.04	0.12	0.13
c2m5 - c2m6	-0.03	0.47	0.98	-0.06	0.23	0.25	-0.03	0.11	0.11
c2m6 - c2m7	0.43	0.48	1.61	0.10	0.24	0.48	0.00	0.12	0.13
c2m7 - c2m8	2.14	0.43	1.35	2.57	0.21	0.44	2.58	0.10	0.11
c2m8 - c2m9	0.04	0.45	0.92	-0.03	0.23	0.25	0.00	0.11	0.12
c2m9 - c2m10	-0.13	0.49	1.15	-0.02	0.24	0.26	0.00	0.12	0.14
c2m10 - c2m11	0.07	0.50	1.21	0.08	0.24	0.28	0.03	0.12	0.14
c2m11 - c2m12	0.10	0.48	1.17	-0.01	0.25	0.33	0.04	0.13	0.13
c2m12 - c2m13	0.16	0.48	1.04	0.05	0.25	0.32	0.01	0.12	0.13
c2m13 - c2m14	-0.09	0.48	0.92	0.08	0.25	0.32	0.02	0.12	0.15
c2m14 - c2m15	0.00	0.48	1.12	-0.05	0.25	0.34	0.06	0.12	0.16
c2m15 - c2m16	0.18	0.50	1.17	0.07	0.25	0.31	0.02	0.12	0.16
c2m16 - c2m17	-0.04	0.49	1.25	0.05	0.24	0.31	0.00	0.12	0.14
c2m17 - c2m18	-0.06	0.48	1.08	0.00	0.24	0.30	0.03	0.12	0.14
c2m18 - c2m19	0.09	0.48	1.06	0.06	0.24	0.31	-0.02	0.12	0.16
c2m19 - c2m20	-0.15	0.49	1.16	0.02	0.24	0.34	0.00	0.12	0.15
c2m20 - c2m21	0.13	0.39	0.95	-0.13	0.19	0.28	0.05	0.10	0.12

Table 7.6: MLE \hat{b}_Q and its estimated standard error from CIM on replicates

Interval	Sample size = 125			Sample size = 500			Sample size = 2000		
	mean MLE \hat{b}_Q	mean imat SD	emp SD	mean MLE \hat{b}_Q	mean imat SD	emp SD	mean MLE \hat{b}_Q	mean imat SD	emp SD
c2m1 - c2m2	-0.06	0.36	0.53	0.06	0.18	0.17	-0.04	0.09	0.10
c2m2 - c2m3	-0.13	0.46	0.78	0.00	0.23	0.28	-0.03	0.11	0.15
c2m3 - c2m4	-0.13	0.45	0.62	-0.06	0.23	0.27	-0.02	0.11	0.12
c2m4 - c2m5	-0.04	0.46	0.81	-0.11	0.23	0.26	-0.01	0.11	0.14
c2m5 - c2m6	-0.04	0.46	0.73	-0.10	0.23	0.23	-0.05	0.11	0.11
c2m6 - c2m7	1.89	0.44	0.71	1.77	0.23	0.32	1.76	0.11	0.15
c2m7 - c2m8	2.66	0.41	0.55	2.62	0.21	0.26	2.58	0.10	0.11
c2m8 - c2m9	0.79	0.45	0.84	0.89	0.23	0.41	0.82	0.11	0.23
c2m9 - c2m10	-0.16	0.46	0.64	-0.07	0.23	0.28	-0.02	0.11	0.15
c2m10 - c2m11	-0.13	0.45	0.72	0.01	0.23	0.29	-0.02	0.11	0.12
c2m11 - c2m12	-0.03	0.46	0.72	-0.07	0.23	0.31	-0.03	0.11	0.12
c2m12 - c2m13	0.01	0.45	0.79	-0.07	0.23	0.26	-0.05	0.11	0.13
c2m13 - c2m14	-0.06	0.45	0.68	-0.01	0.23	0.34	-0.03	0.11	0.15
c2m14 - c2m15	-0.03	0.45	0.75	-0.08	0.23	0.31	-0.01	0.11	0.15
c2m15 - c2m16	0.05	0.46	0.71	0.03	0.23	0.30	-0.04	0.11	0.15
c2m16 - c2m17	-0.10	0.45	0.67	0.05	0.23	0.28	-0.03	0.11	0.15
c2m17 - c2m18	-0.01	0.45	0.78	0.00	0.23	0.29	0.01	0.11	0.15
c2m18 - c2m19	0.08	0.44	0.71	0.00	0.23	0.30	-0.02	0.11	0.15
c2m19 - c2m20	-0.01	0.45	0.77	-0.08	0.23	0.31	-0.01	0.11	0.15
c2m20 - c2m21	0.02	0.36	0.55	-0.24	0.18	0.19	0.01	0.09	0.12

The results presented in Section 7.1.1 revealed that the models may behave differently in intervals that do not contain QTL. Therefore, was necessary to check whether the information matrix gave sensible values for the standard error of QTL effect in other intervals.

Tables 7.5 and 7.6 show, for each interval, the mean of \hat{b}_Q , the mean asymptotic standard error (imat SD), and the empirical standard error (emp SD) generated by RIM1 and CIM respectively. For all intervals sample sizes 500 and 2000, lead to asymptotic standard errors \hat{b}_Q that closely matched the corresponding empirical standard errors. However, at sample size 125 the empirical and asymptotic errors did not agree. This emphasises the fact that the information matrix result rests upon asymptotic theory, so it is not applicable if the sample size is too small.

Is the reliability of the information matrix the same when estimating standard errors of QTL location as when estimating standard errors of QTL effects? The answer depends whether the corresponding QTL effect is close to zero.

Define

$$\text{emp SD of } \hat{p}_{Q2} = \sqrt{\frac{1}{N-1} \left(\sum_{i=1}^N \left(\hat{p}_{Q2}^{(i)} \right)^2 - \frac{1}{N} \left(\sum_{i=1}^N \hat{p}_{Q2}^{(i)} \right)^2 \right)} \quad (7.3)$$

and

$$\text{imat SD of } \hat{p}_{Q2} = \sqrt{[\mathcal{I}(\hat{\psi})]_{p_{Q2} p_{Q2}}^{-1}} \simeq \sqrt{\text{asymptotic var}(\hat{p}_{Q2})}. \quad (7.4)$$

Table 7.7, below, shows that when the testing interval contained a QTL, the asymptotic standard errors of \hat{p}_{Q2} were fairly close to the empirical standard errors. Note that when we look at all intervals, a different picture will be revealed. For RIM1, consider the results in Table 7.8 together with the illustration in Figure 7.10. Likewise, for CIM consider both Table 7.9 and Figure 7.11.

Table 7.7: Simulated Single QTL Case: Interval c2m7-c2m8; comparison of estimated standard errors (SD) of \hat{p}_{Q2} .

Sample size	Model	emp SD of \hat{p}_{Q2}	imat SD of \hat{p}_{Q2}					
125			Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	RIM1	0.358	0.0007	0.001	0.100	0.080	0.135	0.176
	CIM	0.263	0.0007	0.077	0.113	0.097	0.135	0.176
	CIM-QTLcart.	0.273	not applicable					
500			Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	RIM1	0.149	0.0005	0.063	0.069	0.066	0.073	0.083
	CIM	0.117	0.037	0.062	0.069	0.067	0.073	0.083
	CIM-QTLcart.	0.119	not applicable					
2000			Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	RIM1	0.053	0.029	0.032	0.034	0.034	0.035	0.039
	CIM	0.053	0.029	0.032	0.034	0.034	0.035	0.039
	CIM-QTLcart.	0.059	not applicable					

Table 7.8: MLE \hat{p}_{Q2} and its estimated standard error from RIM1 on replicates.

Interval	Sample size = 125			Sample size = 500			Sample size = 2000		
	mean MLE \hat{p}_{Q2}	mean imat SD	emp SD	mean MLE \hat{p}_{Q2}	mean imat SD	emp SD	mean MLE \hat{p}_{Q2}	mean imat SD	emp SD
c2m1 - c2m2	0.61	0.03	0.45	0.73	0.01	0.40	0.60	0.01	0.45
c2m2 - c2m3	0.53	0.04	0.45	0.60	0.01	0.45	0.68	0.01	0.41
c2m3 - c2m4	0.56	0.02	0.46	0.62	0.01	0.45	0.54	0.01	0.47
c2m4 - c2m5	0.59	0.03	0.44	0.52	0.01	0.47	0.58	0.01	0.45
c2m5 - c2m6	0.91	0.02	0.24	0.99	~ 0.00	0.05	~ 1.00	~ 0.00	~ 0.00
c2m6 - c2m7	0.71	0.02	0.42	0.95	~ 0.00	0.20	~ 1.00	~ 0.00	~ 0.00
c2m7 - c2m8	0.51	0.08	0.36	0.63	0.07	0.15	0.68	0.03	0.05
c2m8 - c2m9	0.10	0.02	0.26	~ 0.00	~ 0.00	0.01	~ 0.00	~ 0.00	~ 0.00
c2m9 - c2m10	0.27	0.02	0.42	0.01	~ 0.00	0.11	~ 0.00	~ 0.00	~ 0.00
c2m10 - c2m11	0.42	0.03	0.46	0.40	0.02	0.45	0.39	0.01	0.45
c2m11 - c2m12	0.49	0.03	0.46	0.33	0.01	0.43	0.39	0.01	0.46
c2m12 - c2m13	0.43	0.03	0.46	0.45	0.01	0.47	0.41	0.01	0.45
c2m13 - c2m14	0.45	0.02	0.47	0.48	0.02	0.46	0.45	0.01	0.46
c2m14 - c2m15	0.46	0.03	0.46	0.41	0.01	0.46	0.44	0.01	0.46
c2m15 - c2m16	0.43	0.03	0.45	0.40	0.01	0.45	0.44	0.01	0.46
c2m16 - c2m17	0.53	0.04	0.45	0.52	0.01	0.47	0.42	0.01	0.45
c2m17 - c2m18	0.47	0.03	0.45	0.45	0.02	0.46	0.53	0.01	0.45
c2m18 - c2m19	0.56	0.02	0.47	0.55	0.01	0.47	0.44	0.01	0.45
c2m19 - c2m20	0.42	0.03	0.46	0.46	0.01	0.46	0.40	0.01	0.43
c2m20 - c2m21	0.41	0.03	0.44	0.34	0.03	0.39	0.43	0.01	0.42

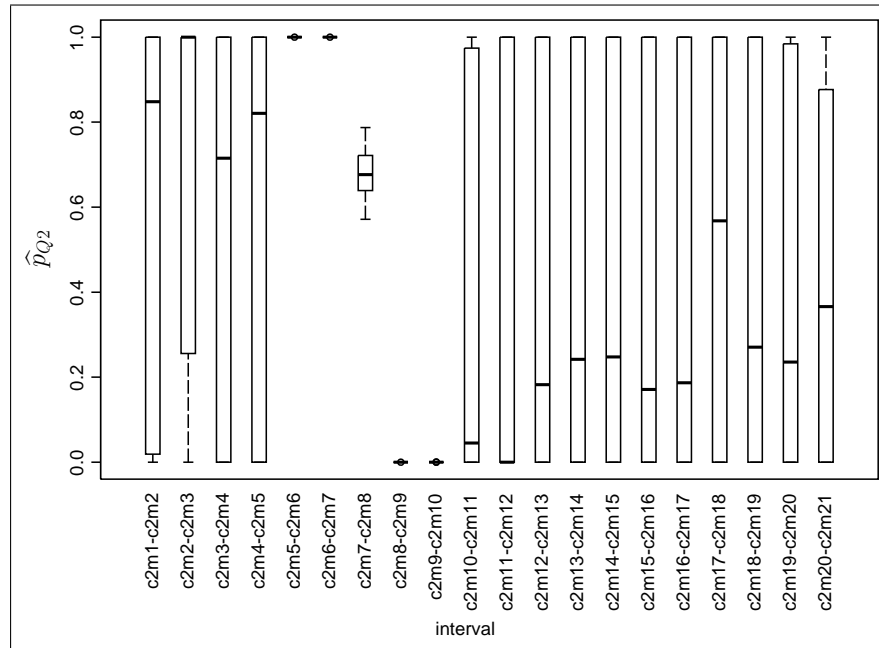
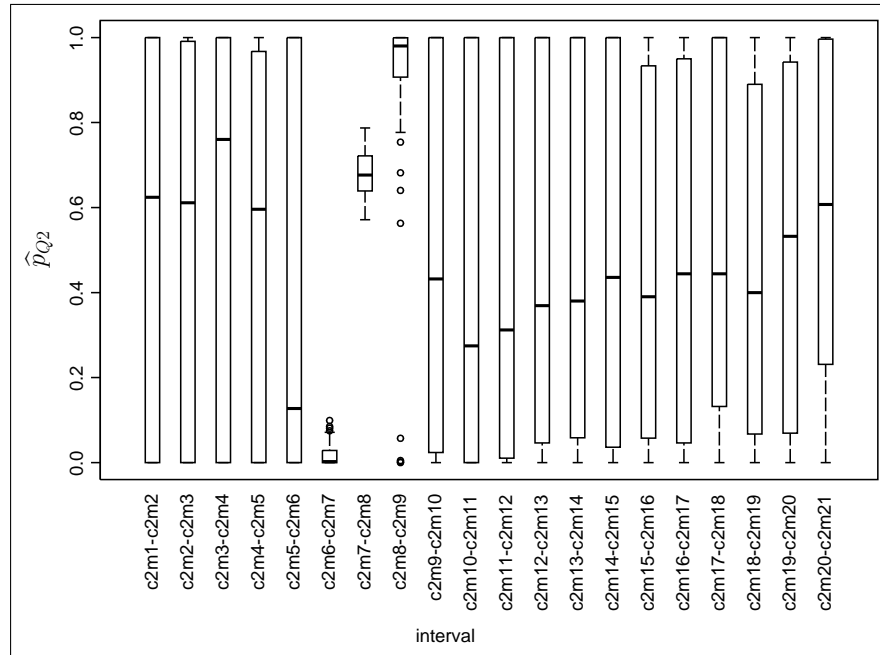
Figure 7.10: Box plots showing distributions of \hat{p}_{Q2} from applying RIM1 to one hundred replicate samples, each having sample size 2000.

Table 7.9: MLE \hat{p}_{Q2} and its estimated standard error from CIM on replicates.

Interval	Sample size = 125			Sample size = 500			Sample size = 2000		
	mean	mean	emp SD	mean	mean	emp SD	mean	mean	emp SD
	MLE \hat{p}_{Q2}	imat SD		MLE \hat{p}_{Q2}	imat SD		MLE \hat{p}_{Q2}	imat SD	
c2m1 - c2m2	0.46	0.03	0.46	0.46	0.02	0.44	0.51	0.01	0.45
c2m2 - c2m3	0.48	0.04	0.45	0.50	0.02	0.44	0.49	0.01	0.44
c2m3 - c2m4	0.46	0.03	0.46	0.44	0.02	0.43	0.57	0.01	0.46
c2m4 - c2m5	0.54	0.03	0.46	0.50	0.02	0.44	0.46	0.01	0.44
c2m5 - c2m6	0.49	0.03	0.46	0.58	0.02	0.44	0.47	~ 0.00	0.48
c2m6 - c2m7	0.13	0.04	0.24	0.05	0.02	0.09	0.02	0.01	0.02
c2m7 - c2m8	0.69	0.10	0.26	0.65	0.07	0.12	0.68	0.03	0.05
c2m8 - c2m9	0.58	0.05	0.42	0.78	0.03	0.33	0.89	0.01	0.24
c2m9 - c2m10	0.51	0.03	0.46	0.53	0.02	0.45	0.54	0.01	0.46
c2m10 - c2m11	0.46	0.04	0.44	0.50	0.02	0.45	0.49	0.01	0.47
c2m11 - c2m12	0.50	0.03	0.45	0.47	0.02	0.43	0.48	0.01	0.45
c2m12 - c2m13	0.50	0.03	0.45	0.59	0.02	0.43	0.51	0.01	0.44
c2m13 - c2m14	0.55	0.02	0.47	0.50	0.03	0.43	0.49	0.01	0.44
c2m14 - c2m15	0.52	0.04	0.45	0.54	0.03	0.43	0.49	0.01	0.42
c2m15 - c2m16	0.51	0.04	0.45	0.57	0.02	0.43	0.46	0.02	0.41
c2m16 - c2m17	0.51	0.04	0.45	0.53	0.03	0.42	0.49	0.01	0.42
c2m17 - c2m18	0.51	0.04	0.45	0.52	0.03	0.42	0.53	0.02	0.40
c2m18 - c2m19	0.63	0.04	0.43	0.44	0.02	0.43	0.46	0.02	0.39
c2m19 - c2m20	0.40	0.04	0.43	0.49	0.03	0.42	0.51	0.02	0.39
c2m20 - c2m21	0.54	0.04	0.45	0.50	0.04	0.39	0.58	0.02	0.38

Figure 7.11: Box plots showing distributions of \hat{p}_{Q2} from applying CIM to one hundred replicate samples, each having sample size 2000.

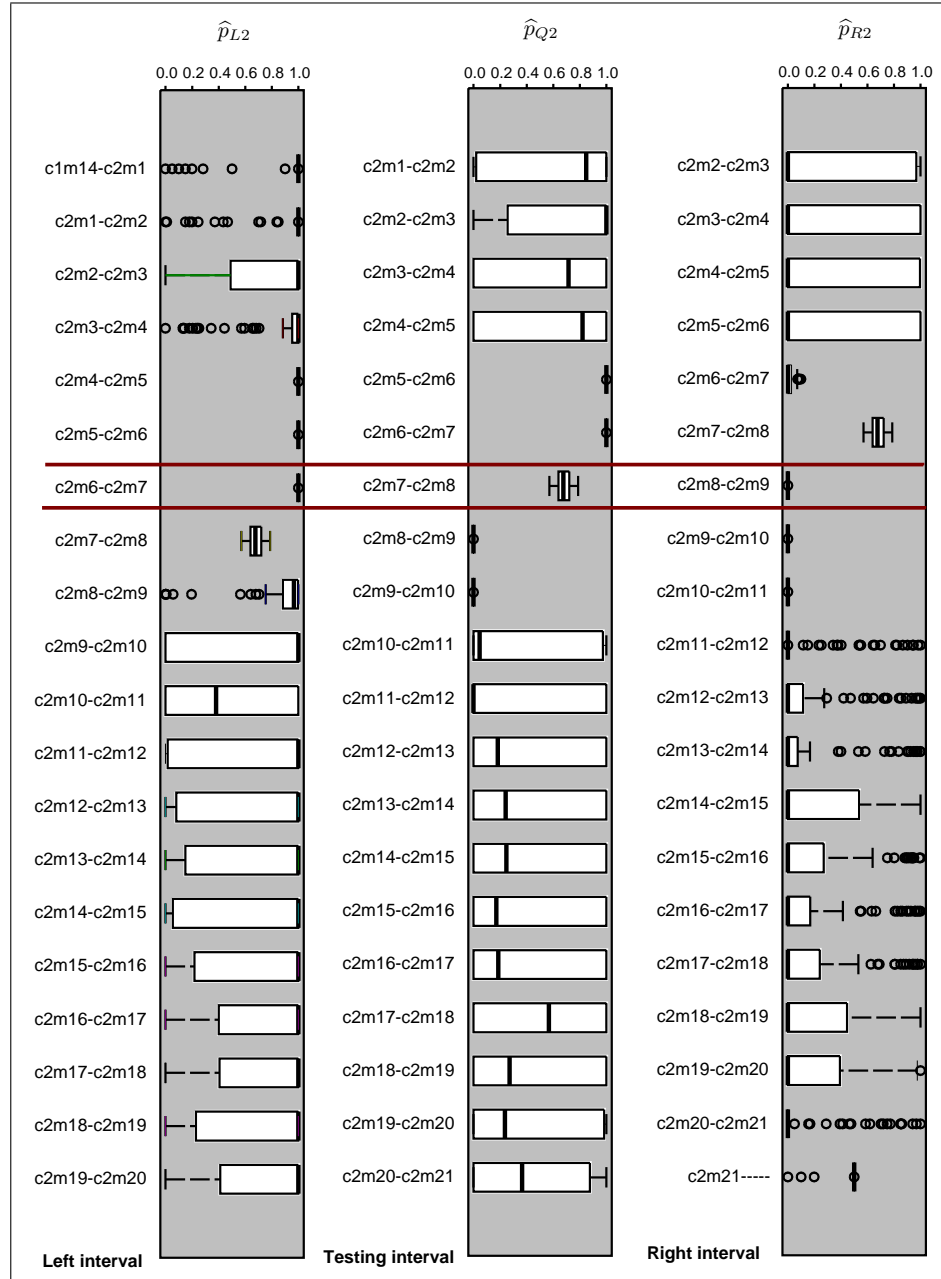


Figure 7.12: Box plots of the estimates \hat{p}_{L2} , \hat{p}_{Q2} and \hat{p}_{R2} obtained by applying RIM1 to 20 consecutive intervals spanning chromosome 2. RIM1 simultaneously fits three QTL (L , Q , R). Each box plot is based on estimates from 100 replicate samples each of size 2000. In the interval with the QTL, the relevant RIM1 estimates are close their true values which are: $p_{R2} = 0.681$ when the testing interval is $c2m6 - c2m7$; $p_{Q2} = 0.681$ when the testing interval is $c2m7 - c2m8$; and $p_{L2} = 0.681$ when the testing interval is $c2m8 - c2m9$.

The Fisher information matrix is block diagonal for mixtures of univariate Normals (see Equation (5.97)). Therefore, it is possible to separately investigate the standard errors of the effects and the mixing parameters.

In Section 5.3, some identifiability problems which plague mixture models were discussed. The impact of loss of identifiability of the mixing parameter p_{Q2} when the effect b_Q is close to zero is revealed in Figures 7.10 and 7.11. In intervals that are located far from a QTL, the estimates of QTL effect (\hat{b}_Q) were close to zero, while the estimates of QTL location (\hat{p}_{Q2}) were unstable, taking on any value within the valid range. Figure 7.12 also shows that the estimates of QTL location are very stable in the interval with the QTL (even when this was not the testing interval), but they were quite unstable elsewhere.

Figures 7.10 and 7.11 give an insight into the behaviour of the EM Algorithm with the RIM1 and CIM models. Figure 7.10 shows that, in the empty intervals adjacent to the interval containing the QTL, RIM1 tends to push the postulated QTL onto the marker that is furthest away from the real QTL (note: $p_{Q2} = 1$ implies that $r_{MQ} = 0$, while $p_{Q2} = 0$ implies that $r_{MQ} = r_{MN}$). If the real QTL is to the right of Q , then locus R is the desired QTL. The EM algorithm in RIM1 acts to reduce the over-specification in RIM1 by pushing Q towards the marker M , and pushing L towards the marker K . Likewise, if the real QTL is L , then RIM1 tends to push Q towards M , and R towards N . Thus the pattern extends over two intervals. While RIM1 models background QTL to absorb the effect of QTLs in the adjacent intervals, CIM does not. Therefore, when applying CIM to an empty interval adjacent to the interval with the real QTL, the EM algorithm tries to adjust for the associated effect by pushing Q as close as it can be to the real QTL (see Figure 7.11).

The impact of loss of identifiability of p_{Q2} when b_Q is close to zero is also evident in Tables 7.8 and 7.9. When a QTL was present in the central testing interval, the information matrix gave good estimates for the standard error of the mixing

parameter p_{Q2} for large samples. However, in intervals where \hat{b}_Q was close to zero, the standard errors for \hat{p}_{Q2} were grossly underestimated by the information matrix.

Comparisons with bootstraps

With real populations, it is not usually feasible to take many replicate samples. In these situations, empirical standard errors of parameter estimates may be via the bootstrap methodology. Therefore it is useful to compare errors obtained from the information matrix with empirical errors obtained using the bootstrap methodology.

One thousand bootstrap samples were generated by re-sampling (with replacement) from the first sample having size 125, 500 and 2000 respectively. A simple, non-parametric bootstrap methodology was used.

At sample size 125, eleven bootstrap samples yielded a singular information matrix. In these cases the Moore-Penrose generalized inverse of the information matrix was used to estimate standard error. At sample sizes 500 and 2000, the bootstraps behaved well because all 1000 bootstraps yielded a non-singular information matrix for every testing interval. The results are summarised in Tables 7.10 and 7.11.

The standard errors from the bootstraps were almost identical to those from the replicates. At sample sizes 500 and 2000, the asymptotic standard errors (imat SD) for b_Q based on the original sample always agreed well with the corresponding bootstrap standard errors.

When the corresponding QTL effect was significantly different from zero, the information matrix estimates for the SD of \hat{p}_{Q2} tended to agree with the corresponding bootstrap standard errors. However, when the QTL effect was zero, the information matrix estimates for the SD of \hat{p}_{Q2} seemed to be infeasibly small.

In intervals located far from QTL, both the replicates and the bootstraps produced empirical standard errors for \hat{p}_{Q2} of between 0.4 and 0.5 (see Tables 7.8, 7.9, and 7.11). The variable p_{Q2} represents a proportion and $\hat{p}_{Q2} \pm 3 \times 0.4$ is always outside its valid

Table 7.10: RIM1 on bootstraps: MLE \hat{b}_Q and its estimated standard error.

Interval	Sample size = 125			Sample size = 500			Sample size = 2000		
	orig MLE \hat{b}_Q	orig imat SD	boot emp SD	orig MLE \hat{b}_Q	orig imat SD	boot emp SD	orig MLE \hat{b}_Q	orig imat SD	boot emp SD
c2m1 - c2m2	-0.40	0.71	1.76	0.20	0.21	0.24	-0.06	0.11	0.12
c2m2 - c2m3	0.58	0.57	2.20	-0.24	0.24	0.34	0.16	0.11	0.19
c2m3 - c2m4	0.50	0.58	1.86	0.12	0.26	0.32	-0.02	0.12	0.18
c2m4 - c2m5	-0.30	0.48	1.53	0.29	0.26	0.50	-0.03	0.12	0.13
c2m5 - c2m6	0.28	0.57	1.39	0.37	0.24	0.29	-0.11	0.11	0.12
c2m6 - c2m7	-0.44	0.60	2.14	0.09	0.25	0.33	0.04	0.12	0.14
c2m7 - c2m8	3.20	0.50	2.19	2.67	0.21	0.29	2.74	0.10	0.11
c2m8 - c2m9	0.17	0.44	1.12	-0.42	0.24	0.30	-0.11	0.11	0.12
c2m9 - c2m10	-0.92	0.60	1.43	0.16	0.21	0.28	-0.17	0.13	0.15
c2m10 - c2m11	-0.24	0.51	1.46	0.47	0.21	0.25	0.19	0.11	0.24
c2m11 - c2m12	-0.69	0.47	1.70	-0.23	0.24	0.31	0.28	0.13	0.24
c2m12 - c2m13	-0.71	0.50	1.49	-0.03	0.25	0.33	0.05	0.13	0.19
c2m13 - c2m14	0.24	0.42	1.33	-0.02	0.25	0.35	0.23	0.11	0.18
c2m14 - c2m15	0.86	0.43	1.74	-0.23	0.24	0.36	-0.14	0.12	0.20
c2m15 - c2m16	-0.84	0.50	1.73	0.46	0.22	0.35	-0.01	0.13	0.16
c2m16 - c2m17	-0.81	0.44	1.51	0.00	0.24	0.36	-0.01	0.12	0.15
c2m17 - c2m18	-0.12	0.46	1.66	-0.25	0.24	0.43	0.09	0.13	0.15
c2m18 - c2m19	0.75	0.47	1.76	0.12	0.24	0.43	-0.08	0.13	0.15
c2m19 - c2m20	-1.88	0.66	1.82	0.13	0.25	0.47	0.06	0.14	0.15
c2m20 - c2m21	2.02	0.59	1.71	-0.27	0.19	0.35	0.01	0.08	0.13

Table 7.11: RIM1 on bootstraps: MLE \hat{p}_{Q2} and its estimated standard error.

Interval	Sample size = 125			Sample size = 500			Sample size = 2000		
	orig MLE \hat{p}_{Q2}	orig imat SD	boot emp SD	orig MLE \hat{p}_{Q2}	orig imat SD	boot emp SD	orig MLE \hat{p}_{Q2}	orig imat SD	boot emp SD
c2m1 - c2m2	~ 0.00	~ 0.00	0.46	~ 1.00	~ 0.00	0.32	~ 1.00	~ 0.00	0.29
c2m2 - c2m3	0.02	0.02	0.41	~ 1.00	~ 0.00	0.42	0.19	0.03	0.43
c2m3 - c2m4	~ 1.00	~ 0.00	0.45	~ 1.00	~ 0.00	0.35	~ 0.00	~ 0.00	0.43
c2m4 - c2m5	0.97	0.04	0.46	0.01	0.02	0.46	~ 1.00	~ 0.00	0.42
c2m5 - c2m6	~ 1.00	~ 0.00	0.29	~ 1.00	~ 0.00	0.04	~ 1.00	~ 0.00	~ 0.00
c2m6 - c2m7	~ 1.00	~ 0.00	0.47	~ 1.00	~ 0.00	0.09	~ 1.00	~ 0.00	~ 0.00
c2m7 - c2m8	0.71	0.14	0.31	0.61	0.08	0.13	0.67	0.03	0.06
c2m8 - c2m9	~ 0.00	~ 0.00	0.19	~ 0.00	~ 0.00	0.03	~ 0.00	~ 0.00	~ 0.00
c2m9 - c2m10	~ 0.00	~ 0.00	0.39	~ 0.00	~ 0.00	0.22	~ 0.00	~ 0.00	~ 0.00
c2m10 - c2m11	~ 1.00	~ 0.00	0.49	0.64	0.06	0.37	0.06	0.02	0.42
c2m11 - c2m12	~ 0.00	0.02	0.43	~ 0.00	~ 0.00	0.31	~ 1.00	~ 0.00	0.40
c2m12 - c2m13	~ 1.00	~ 0.00	0.49	~ 0.00	~ 0.00	0.47	~ 0.00	~ 0.00	0.48
c2m13 - c2m14	~ 1.00	~ 0.00	0.46	~ 1.00	~ 0.00	0.44	0.31	0.03	0.37
c2m14 - c2m15	~ 0.00	~ 0.00	0.46	~ 1.00	~ 0.00	0.47	0.00	~ 0.00	0.43
c2m15 - c2m16	~ 0.00	~ 0.00	0.47	0.65	0.07	0.39	~ 0.00	~ 0.00	0.41
c2m16 - c2m17	0.98	0.04	0.46	~ 0.00	~ 0.00	0.44	~ 1.00	~ 0.00	0.44
c2m17 - c2m18	~ 1.00	~ 0.00	0.43	0.00	~ 0.00	0.43	1.00	~ 0.00	0.45
c2m18 - c2m19	0.10	0.08	0.45	~ 0.00	~ 0.00	0.44	~ 0.00	~ 0.00	0.40
c2m19 - c2m20	~ 0.00	~ 0.00	0.41	~ 1.00	~ 0.00	0.46	~ 0.00	~ 0.00	0.45
c2m20 - c2m21	0.32	0.16	0.33	0.92	0.04	0.44	~ 0.00	~ 0.00	0.41

range. Therefore, when the QTL effect is zero, it is inappropriate to use the empirical standard error of \hat{p}_{Q2} together with assumptions of Normality. This does not put a caveat on proceeding because estimates of QTL location (and their standard errors) are only of interest when the corresponding QTL effects are significantly different from zero.

The information matrix formula given in Equation (5.97) is a reliable method for estimating the standard error of MLEs in Normal mixture model, and its reliability improves with increasing sample size. For sufficiently large samples, the information matrix appears to be particularly good at estimating the standard error associated with component means (for example QTL effects). However, if one or more component means are close to zero, the information matrix can yield poor estimates for certain mixing proportions.

7.1.3 Hypothesis testing

Formal hypothesis testing was used to investigate the abilities of RIM1, CIM and simple interval mapping (IM) to avoid ghosting and to detect the isolated QTL.

IM and CIM output from QTL Cartographer were tested using the likelihood ratio test and its usual chi-squared distribution with degrees of freedom equal to the difference between the number of parameters under the null and alternate hypotheses.

RIM1 and CIM output from our bespoke implementations were tested using the statistics T_1 and J_1 as defined in equations (5.98) to (5.100). The statistics T_1 and J_1 were calculated using asymptotic standard errors as given in equations (7.2) and (7.4) respectively. The permutation method, described in Chapter 5, was also used for hypothesis testing with test statistics T_2 and J_2 . Tables 7.12 and 7.13 display the results.

Table 7.12: Percent of times p-value < 0.001 for ten testing methods. Tests applied to single-QTL situation; 100 replicate B1 backcross samples each of size $n = 2000$; data simulated using QTL Cartographer. The LRT results are from QTL Cartographer. The statistics T_1 , and T_2 are based on \hat{b}_Q only, while the statistics J_1 and J_2 are used to construct joint tests for \hat{b}_Q and \hat{p}_{Q2} . The columns marked ‘asy’ are based on an asymptotic null distribution, and those marked ‘emp’ are based on an empirical null distribution obtained by the permutation method.

Testing Interval	IM	CIM					RIM1				True QTL
	asy LRT	asy LRT	asy T_1	emp T_2	asy J_1	emp J_2	asy T_1	emp T_2	asy J_1	emp J_2	
c2m1 - c2m2	100	1	1	0	0	0	1	1	0	0	-
c2m2 - c2m3	100	1	2	1	2	1	2	1	2	1	-
c2m3 - c2m4	100	0	1	0	1	0	1	0	1	0	-
c2m4 - c2m5	100	0	1	0	1	1	2	1	1	1	-
c2m5 - c2m6	100	0	2	0	2	1	0	0	0	0	-
c2m6 - c2m7	100	100	100	100	15	22	0	0	0	0	-
c2m7 - c2m8	100	100	100	100	100	100	100	100	100	100	QTL 9
c2m8 - c2m9	100	100	94	92	31	19	0	0	0	0	-
c2m9 - c2m10	100	1	4	0	3	0	1	0	0	2	-
c2m10 - c2m11	100	0	1	0	1	0	0	0	0	0	-
c2m11 - c2m12	100	0	1	0	1	0	1	0	1	0	-
c2m12 - c2m13	100	0	0	0	0	0	0	0	0	0	-
c2m13 - c2m14	100	0	0	0	0	0	0	0	0	0	-
c2m14 - c2m15	100	0	0	0	0	0	1	1	0	0	-
c2m15 - c2m16	100	1	3	2	3	1	4	3	3	2	-
c2m16 - c2m17	98	0	0	0	0	0	0	0	0	0	-
c2m17 - c2m18	81	0	0	0	0	0	0	0	0	0	-
c2m18 - c2m19	57	0	0	0	0	0	0	0	0	0	-
c2m19 - c2m20	33	1	1	0	1	0	1	0	1	0	-
c2m20 - c2m21	15	0	0	1	0	0	0	0	0	0	-

Table 7.13: Percent of times p-value < 0.001 for ten testing methods. Tests applied to single-QTL situation; data simulated using QTL Cartographer; sample sizes 500 and 125. The LRT results are from QTL Cartographer. The statistics T_1 , and T_2 are based on \hat{b}_Q only, while the statistics J_1 and J_2 are used to construct joint tests for \hat{b}_Q and \hat{p}_{Q2} . The columns marked ‘asy’ are based on an asymptotic null distribution, and those marked ‘emp’ are based on an empirical null distribution obtained by the permutation method.

(a) Backcross sample size $n = 500$ (single-QTL).

Testing Interval	IM	CIM						RIM1				True QTL
	asy LRT	asy LRT	asy T_1	emp T_2	asy J_1	emp J_2	asy T_1	emp T_2	asy J_1	emp J_2		
c2m1 - c2m2	100	0	0	0	0	0	0	0	0	0	-	
c2m2 - c2m3	100	0	0	0	0	0	0	0	0	0	-	
c2m3 - c2m4	100	1	1	0	0	0	1	0	0	0	-	
c2m4 - c2m5	100	0	0	0	0	0	0	0	0	0	-	
c2m5 - c2m6	100	0	0	0	0	0	0	1	0	1	-	
c2m6 - c2m7	100	100	100	95	12	15	4	0	1	0	-	
c2m7 - c2m8	100	100	100	100	96	99	98	5	95	13	QTL 9	
c2m8 - c2m9	100	66	72	35	21	6	0	0	0	0	-	
c2m9 - c2m10	100	0	0	0	0	0	0	0	0	2	-	
c2m10 - c2m11	100	0	1	0	1	0	0	0	0	0	-	
c2m11 - c2m12	100	1	2	2	1	0	2	1	2	2	-	
c2m12 - c2m13	99	1	1	1	0	0	1	1	0	0	-	
c2m13 - c2m14	97	0	1	0	0	0	1	0	0	0	-	
c2m14 - c2m15	87	0	1	0	0	0	2	0	0	0	-	
c2m15 - c2m16	47	0	1	0	1	0	2	0	1	0	-	
c2m16 - c2m17	23	1	1	1	1	1	1	1	1	1	-	
c2m17 - c2m18	7	0	0	0	0	0	0	0	0	0	-	
c2m18 - c2m19	3	0	0	0	0	0	0	0	0	0	-	
c2m19 - c2m20	0	0	1	0	1	0	1	0	1	0	-	
c2m20 - c2m21	0	1	2	0	2	0	2	0	2	0	-	

(b) Backcross sample size $n = 125$ (single-QTL).

Testing Interval	IM	CIM						RIM1				True QTL
	asy LRT	asy LRT	asy T_1	emp T_2	asy J_1	emp J_2	asy T_1	emp T_2	asy J_1	emp J_2		
c2m1 - c2m2	29	0	1	0	1	0	9	0	1	0	-	
c2m2 - c2m3	49	3	6	0	0	0	16	0	1	0	-	
c2m3 - c2m4	84	1	2	0	0	0	13	0	2	0	-	
c2m4 - c2m5	98	1	7	1	0	1	23	0	3	0	-	
c2m5 - c2m6	100	2	6	0	0	0	12	1	2	0	-	
c2m6 - c2m7	100	66	76	0	1	0	32	1	1	0	-	
c2m7 - c2m8	100	95	98	14	12	18	74	0	9	1	QTL 9	
c2m8 - c2m9	100	18	20	6	1	0	11	0	0	0	-	
c2m9 - c2m10	100	0	1	0	0	0	9	0	0	0	-	
c2m10 - c2m11	89	2	6	0	1	1	10	2	1	2	-	
c2m11 - c2m12	62	1	4	0	2	0	13	1	2	2	-	
c2m12 - c2m13	38	2	5	1	2	0	10	0	3	1	-	
c2m13 - c2m14	19	0	1	0	0	0	5	1	1	0	-	
c2m14 - c2m15	7	0	6	0	1	0	16	0	1	0	-	
c2m15 - c2m16	4	2	5	0	1	0	13	1	2	0	-	
c2m16 - c2m17	2	0	3	0	0	0	16	0	5	0	-	
c2m17 - c2m18	2	2	4	2	1	1	11	2	1	1	-	
c2m18 - c2m19	2	2	3	0	0	0	11	2	0	0	-	
c2m19 - c2m20	1	3	4	0	2	0	11	0	1	0	-	
c2m20 - c2m21	1	1	3	0	0	0	13	0	0	1	-	

Simple interval mapping (IM) with the LRT was able to detect QTL but exhibited severe ghosting everywhere. As pointed out by Zeng (1994), the fitting of extra markers as cofactors in CIM helps to absorb background QTL effect thus enabling better determination of QTL location with CIM than with IM. RIM1 exploits the strengths of the CIM model by also fitting marker cofactors, and it adds putative QTL in interval adjacent to the testing interval in order to reduce ghosting.

Tests based on the LRT, and tests based on T_1 and J_1 , are all rough tests in the sense that all of their assumed asymptotic properties may not fully hold for Normal mixture models. Nevertheless, the results show that they can yield informative results because they all have power to detect QTL. At sample size 2000, there was good agreement between the asymptotic tests based on T_1 and J_1 respectively and the corresponding permutation tests (based on T_2 and J_2).

The tests CIM (LRT) and CIM T_1 tended to give similar results with strong (98%-100%) power to detect the isolated QTL, severe ghosting in intervals adjacent to the QTL and with little ghosting in intervals further away. The joint test J_1 dramatically reduced ghosting in CIM while retaining its power to detect QTL.

RIM1 out-performed CIM at sample sizes 500 and 2000 with virtually no ghosting from tests T_1 and J_1 and power of 95% to 100%. While the joint test seemed to be essential for reducing ghosting in CIM, it may be noted that the joint test was not essential for reducing ghosting in RIM1. The simple test for QTL effect was enough to reduce ghosting in RIM1. This indicates that the form of the RIM1 model is robust against ghosting.

The tests CIM (LRT) and CIM T_1 exhibited more power to detect QTL and more stability at sample size 125 than the test RIM1 T_1 . However at this small sample size ghosting all models experienced more false detections in intervals located far away from the QTL. At sample size 125, the joint test J_1 experienced almost complete loss of power to detect QTL for both RIM1 and CIM. When we take another look

at Figures 7.2 and 7.3, these results make sense. At sample size 125, the confidence intervals for QTL location tended to stretch across the entire testing interval. The empirical tests based on T_2 and J_2 also performed poorly at sample size 125. At small sample size, the permutation tests were more sensitive to the significance level than the asymptotic tests. Figure 7.13 illustrates that for small samples, the permutations did not correctly represent the null hypothesis.

Sometimes, with real data there may be no option but to work with small sample sizes. A QTL with high heritability or large effects may be easy detect and locate with small sample sizes. However, one has to acknowledge that in most situations, it may be unrealistic to expect to precisely estimate QTL location using very small samples. If the sample size is extremely small it might be necessary to concentrate on QTL detection and to de-emphasize the desire to find precise location.

With very small sample sizes it is helpful to select models having few parameters. For example, one might consider fitting CIM instead of RIM1 or one might consider fitting fewer cofactors. In our simulations, all available markers were used as cofactors (that is, all markers except the ones that we conditioned on). This would not be an ideal strategy when working with small sample sizes. It would be better use stepwise regression or similar techniques to select a small subset of the available markers to include as cofactors. There is also the problem of marker spacing. If map density is high, then small sample sizes may not offer much chance to detect recombination between marker and QTL as there may be too few recombinant individuals within the sample. Therefore, for small sample sizes one might also consider using more widely spaced markers.

None of the interval mapping models explored in this chapter included interactions between QTL. Chapter 8 discusses how to include such interactions. However, it must be noted that if the sample size is too small, it may not be beneficial to add extra interaction terms to a CIM or RIM1 model.

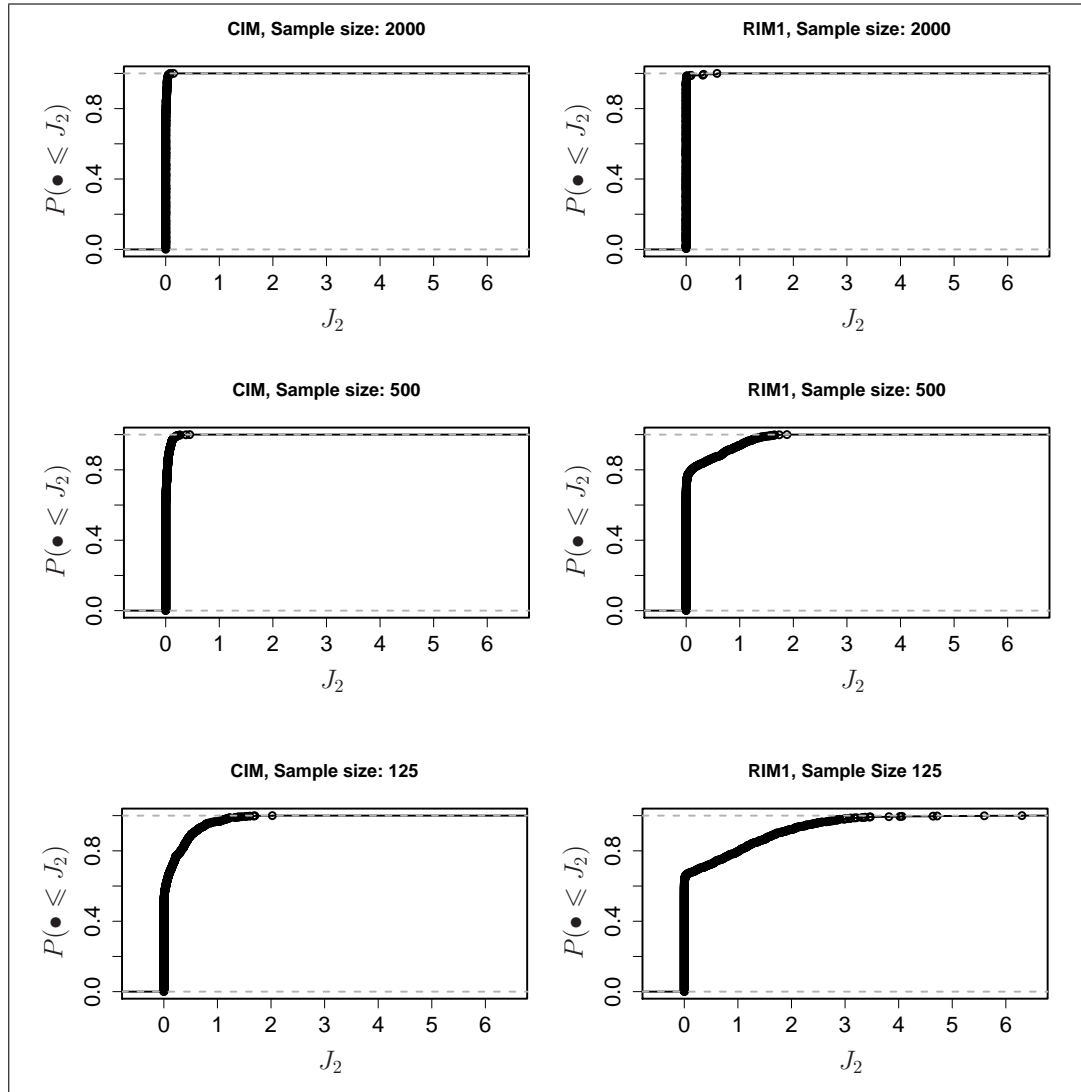


Figure 7.13: The empirical distribution of $J_2 = \hat{p}_{Q2}(1 - \hat{p}_{Q2})(\hat{b}_Q)^2$ for interval $c2m7 - c2m8$ based on 1000 permutations. The original samples were the first replicate at each sample size. These three original samples had a single QTL in $c2m7 - c2m8$ with $J_2 = 1.446$. The permutations were designed to remove the effect of this QTL. If the permutations correctly generated data under H_0 , then J_2 should always be close to zero. At sizes 500 and 125, the permutations gave more stable results with CIM than with RIM1. The plots show that the permutations worked best when the original sample size was large.

7.2 The Multi-QTL Situation

It is useful to examine the behaviour of RIM1 and CIM and the performance of our proposed hypothesis test in a situation where the trait is controlled by multiple QTL.

The QTL Cartographer module Rcross was used to simulate B1 backcross samples in the multiple QTL situation. In Rcross, Haldane's map function was assumed and the trait values were determined using the Cockerham (1954) genetic model.

The simulations were based on the same marker-map that was used in our single-QTL situation. The genetic map in Figure 7.1 was modified by adding ten extra QTL (in addition to the QTL named *QTL 9*), giving eleven QTL altogether. Figure 7.14 shows the resulting genetic map. The QTL locations were chosen in an *ad hoc* manner, but the aim was to have QTL with a variety of sizes and directions of effects. Likewise, the QTL locations were chosen to make QTL detection potentially difficult. For example, one QTL was made to coincide with a marker and two QTL were placed in adjacent intervals. Table 7.14 lists the full specification of QTL effects and locations used for these simulations. The heritability of the trait was set to $1/2$, causing the error variance (σ^2) to be equal to overall genetic variance. The expected value of σ^2 was approximately equal to 10.22 for these samples.

Table 7.14: Multi-QTL case: QTL locations and effects used for simulations. The parameters d_0 and a_0 were input into QTL cartographer and QTL Cartographer calculated the genotypic values of QQ , Qq and qq as $u_{QQ} = a_0 - \frac{1}{2}d_0$, $u_{Qq} = -\frac{1}{2}d_0$ and $u_{qq} = -a_0 - \frac{1}{2}d_0$, respectively.

<i>Q</i>	chromosome	<i>M</i>	<i>N</i>	r_{MQ}	r_{QN}	Additive $a_0 = a_{QQ}$	Dominance $d_0 = -2d_{QQ}$	$b_Q = (\mu_{QQ} - \mu_{Qq})$ $= (a_0 - d_0)$
<i>QTL 1</i>	1	c1m7	c1m8	0.0305	0.0641	2.52	0.20	2.32
<i>QTL 2</i>	2	c2m11	c2m12	0.0476	0.0476	0.99	-0.38	1.37
<i>QTL 3</i>	2	c2m15	c2m16	0.0668	0.0275	-1.38	-0.47	1.85
<i>QTL 4</i>	1	c1m1	c1m2	0.0574	0.0376	0.60	-2.41	3.01
<i>QTL 5</i>	1	c1m13	c1m14	0.0396	0.0554	0.48	0.82	-0.34
<i>QTL 6</i>	2	c2m4	c2m5	0.0436	0.0515	0.71	0.14	0.57
<i>QTL 7</i>	1	c1m5	c1m6	0.0569	0.0380	0.45	0.45	0.00
<i>QTL 8</i>	2	c2m3	c2m4	0.0781	0.0148	0.92	-0.37	1.29
<i>QTL 9</i>	2	c2m7	c2m8	0.0309	0.0637	3.14	0.56	2.58
<i>QTL 10</i>	2	c2m19	c2m20	0.0000	0.0906	0.70	0.39	0.31
<i>QTL 11</i>	2	c2m13	c2m14	0.0668	0.0275	1.38	0.47	0.91

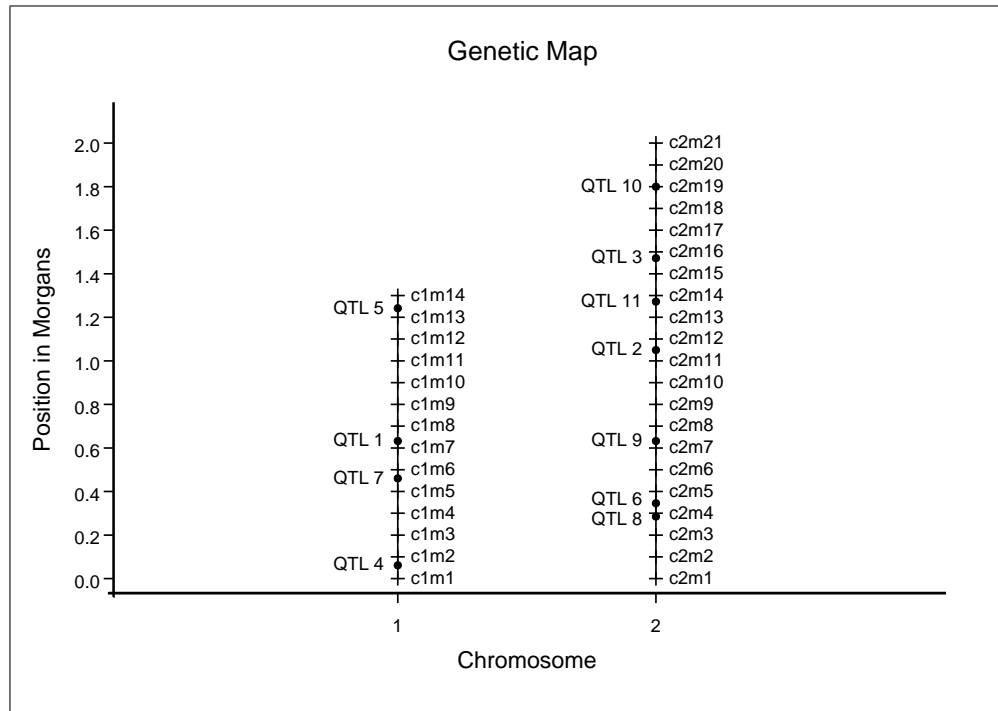


Figure 7.14: Multi-QTL, genetic map on which simulations were based.

The main aim was to scan chromosome two in search of QTL. Another aim was to examine how detection of *QTL 9* (an isolated QTL with good-sized effects) is affected by the presence of the other QTL. The results of hypothesis testing based on CIM and RIM1 are presented in Table 7.15, for sample sizes five hundred and two thousand.

At sample size 2000, all models had good power to detect *QTL 9*. In the single-QTL case, moving from sample size 2000 to 500 caused a drop in power of around five percentage points. In the multi-QTL case, moving from sample size 2000 to 500 caused a drop in power of around 20 percentage points. This indicates that if a trait is controlled by multiple QTL then a larger sample size may be required for detection than when only one QTL is involved. The tests CIM (LRT) and CIM T_1 showed highest rate of detection and the highest false positive error rates.

Table 7.15: Percent of times p-value < 0.001 for nine testing methods. Tests applied to multi-QTL situation; 100 replicate B1 backcross samples for $n = 2000, 500$ and 125; data simulated using QTL Cartographer. The LRT results are from QTL Cartographer. The statistics T_1 , and T_2 are based on \hat{b}_Q only, while the statistics J_1 and J_2 are used to construct joint tests for \hat{b}_Q and \hat{p}_{Q2} . The columns marked ‘asy’ are based on an asymptotic null distribution, and those marked ‘emp’ are based on an empirical null distribution obtained by the permutation method.

(a) CIM and RIM1 with $n = 2000$ and multiple QTL.

Testing Interval	CIM					RIM1				True QTL
	asy LRT	asy T_1	emp T_2	asy J_1	emp J_2	asy T_1	emp T_2	asy J_1	emp J_2	
c2m1 - c2m2	2	2	0	0	0	2	1	0	0	-
c2m2 - c2m3	1	2	2	2	1	0	0	0	0	-
c2m3 - c2m4	100	100	96	62	33	38	38	31	38	QTL 8
c2m4 - c2m5	99	99	98	74	60	54	53	49	48	QTL 6
c2m5 - c2m6	2	3	0	3	1	1	0	0	0	-
c2m6 - c2m7	100	98	96	40	24	5	5	5	5	-
c2m7 - c2m8	100	100	100	97	97	93	93	90	92	QTL 9
c2m8 - c2m9	41	46	38	31	12	0	0	0	0	-
c2m9 - c2m10	1	4	0	4	0	2	0	1	0	-
c2m10 - c2m11	25	36	13	28	3	7	4	5	6	-
c2m11 - c2m12	88	97	77	96	62	68	61	66	52	QTL 2
c2m12 - c2m13	38	51	36	47	31	5	4	4	5	-
c2m13 - c2m14	49	60	33	52	19	29	18	16	19	QTL 11
c2m14 - c2m15	14	13	11	7	1	17	17	3	16	-
c2m15 - c2m16	47	58	42	50	24	34	24	26	14	QTL 3
c2m16 - c2m17	28	35	8	23	5	23	5	21	5	-
c2m17 - c2m18	0	0	0	0	0	0	0	0	0	-
c2m18 - c2m19	1	6	0	5	0	5	0	5	0	c2m19 = QTL 10
c2m19 - c2m20	2	7	0	5	0	6	0	5	0	c2m19 = QTL 10
c2m20 - c2m21	0	2	0	2	0	2	0	2	0	-

(b) CIM and RIM1 with $n = 500$ and multiple QTL.

Testing Interval	CIM					RIM1				True QTL
	asy LRT	asy T_1	emp T_2	asy J_1	emp J_2	asy T_1	emp T_2	asy J_1	emp J_2	
c2m1 - c2m2	0	0	0	0	0	0	0	0	0	-
c2m2 - c2m3	0	1	0	1	0	0	0	0	0	-
c2m3 - c2m4	23	38	20	21	10	32	12	19	9	QTL 8
c2m4 - c2m5	27	42	14	23	8	32	4	19	8	QTL 6
c2m5 - c2m6	0	0	0	0	0	3	3	0	2	-
c2m6 - c2m7	41	49	16	24	5	14	1	10	15	-
c2m7 - c2m8	84	92	74	71	62	78	32	61	40	QTL 9
c2m8 - c2m9	7	8	3	2	1	0	0	0	2	-
c2m9 - c2m10	0	1	0	1	1	1	0	1	1	-
c2m10 - c2m11	5	8	2	3	2	7	3	3	2	-
c2m11 - c2m12	13	27	18	23	11	24	8	22	11	QTL 2
c2m12 - c2m13	5	12	2	9	7	8	3	6	5	-
c2m13 - c2m14	11	19	3	13	5	15	3	10	4	QTL 11
c2m14 - c2m15	0	1	2	1	0	5	5	2	0	-
c2m15 - c2m16	3	4	3	2	1	4	3	2	1	QTL 3
c2m16 - c2m17	1	3	2	1	1	3	2	1	1	-
c2m17 - c2m18	0	0	0	0	0	0	0	0	0	-
c2m18 - c2m19	1	5	0	4	0	5	0	4	0	c2m19 = QTL 10
c2m19 - c2m20	0	1	0	1	0	2	0	1	0	c2m19 = QTL 10
c2m20 - c2m21	1	2	0	2	0	2	0	2	0	-

Table 7.15: Continued: percent of times p-value < 0.001 for nine testing methods.

(c) CIM and RIM1 with $n = 125$ and multiple QTL.

Testing Interval	CIM					RIM1				True QTL
	asy LRT	asy T_1	emp T_2	asy J_1	emp J_2	asy T_1	emp T_2	asy J_1	emp J_2	
c2m1 - c2m2	0	1	0	1	0	11	0	2	0	-
c2m2 - c2m3	2	3	0	0	0	14	0	1	0	-
c2m3 - c2m4	1	8	0	0	0	16	0	0	1	QTL 8
c2m4 - c2m5	9	16	2	2	2	21	1	3	1	QTL 6
c2m5 - c2m6	1	5	0	0	0	15	0	1	2	-
c2m6 - c2m7	8	15	0	1	0	22	1	3	0	-
c2m7 - c2m8	15	27	2	4	1	29	0	4	0	QTL 9
c2m8 - c2m9	5	8	0	0	0	13	0	1	0	-
c2m9 - c2m10	0	1	0	0	0	9	0	0	0	-
c2m10 - c2m11	1	6	0	1	0	10	1	1	1	-
c2m11 - c2m12	2	9	0	0	0	13	1	0	1	QTL 2
c2m12 - c2m13	4	7	1	2	1	11	4	2	2	-
c2m13 - c2m14	2	5	0	1	0	10	0	1	0	QTL 11
c2m14 - c2m15	1	4	0	0	0	15	0	0	0	-
c2m15 - c2m16	3	5	0	1	0	12	1	2	0	QTL 3
c2m16 - c2m17	1	9	0	1	0	18	0	2	0	-
c2m17 - c2m18	0	4	1	0	2	13	2	1	0	-
c2m18 - c2m19	2	4	0	0	0	11	1	0	0	c2m19 = QTL 10
c2m19 - c2m20	2	4	0	1	0	14	1	2	0	c2m19 = QTL 10
c2m20 - c2m21	1	3	0	1	0	15	1	2	1	-

In general, the power to detect QTL was lower in the multi-QTL situation than in the single-QTL situation. None of the methods detected *QTL 10*. However the lack of detection of *QTL 10* may not be due to the fact that this QTL lies on a marker. It is most likely due to the fact that while the background error was large ($\sigma^2 = 10.22$), *QTL 10* had negligible effects with $b_Q = 0.31$. It is certainly possible to detect QTL that lie on a marker. For example, Table 8.4 shows a real-data situation in which several QTL were found to coincide with markers.

Although chromosome 1 was not scanned for QTL, it is interesting to note that *QTL 7*, which has $b_Q = 0$, could not have been detected using a backcross design. The next chapter discusses how RIM1 may be applied to other breeding designs.

Chapter 8

Other Breeding Designs and Real Data Applications

8.1 Including interactions between QTL

Working with contrast matrices allows us to easily specify a variety of linear models to explain how genotypes at different loci combine to determine trait value. To include interaction effects, we choose appropriate contrasts of the QTL genotypic means to extract the desired effects. These contrasts become additional columns of \mathbf{C} and must be chosen such that the rank \mathbf{C} is equal to the number of columns of \mathbf{C} . For example, to include all possible interactions in the backcross model, we define a new contrast matrix \mathbf{C} of the form

$$\mathbf{C} = (\mathbf{C}_{\bullet 1}, \dots, \mathbf{C}_{\bullet 4}, \mathbf{C}_{\bullet 5}, \dots, \mathbf{C}_{\bullet t}),$$

where $(t - 4)$ is the number of interaction effects being fitted, and $\mathbf{C}_{\bullet 1}, \dots, \mathbf{C}_{\bullet 4}$ are the same as in Equation (5.3). We also have additional elements in the parameter \mathbf{b} , so that

$$\mathbf{b} = (b_0, \dots, b_3, b_4, \dots, b_{t-1}).$$

The rest of the model remains unchanged.

8.2 Application to Other Inbred Designs

The backcross and F2, as well as other breeding designs may be implemented in RIM1 simply by specifying the matrix of category identities (\mathbf{Z}), the QTL contrast matrix (\mathbf{C}), the parameters associated with genotypic effects ($\boldsymbol{\beta}$) and the matrix of conditional QTL genotype probabilities (\mathbf{W}). The rest of the machinery remains unaltered. Section 8.2.1 illustrates the details of the implementation of RIM1 for F2 linecross data.

The backcross and F2 designs involve two alleles at each locus and so our discussions, thus far, have been restricted to bi-allelic loci. However, the RIM1 model also applies to inbred designs where loci can have more than two alleles. Such designs are often used in practice. For example, the mouse consortium work directed by Churchill uses eight-way recombinant inbred (RI) lines created from eight commonly used mouse strains (see Williams *et al.* 2002). These eight-way RI strains exhibit a mix of alleles at each locus. Section 8.2.2 explains how RIM1 may be adapted to suit the situation of multiple alleles at each locus.

8.2.1 Application to the F2

Let us first configure the matrix of conditional QTL genotype probabilities to reflect the properties of the F2 design. The definitions of the conditional F1 transmission probabilities p_{L1} , p_{L2} , p_{Q1} , p_{Q2} and p_{R1} , p_{R2} remain as given in Equations (5.8) to (5.13).

We now need to determine how these transmission probabilities combine to form conditional genotype probabilities in the F2. The easiest way to do this is to work with the recombination probabilities π_{00} , π_{01} , π_{10} and π_{11} , which are defined in Equations (2.1) to (2.4). For example, if an F2 individual has genotype $MMQQNN$ then both F1 parents had to transmit the haplotype MQN . The probability that an F1 parent transmits MQN is equal to $\pi_{00}/2$. Therefore, $\pi_{00}/4$ represents the probability that an F2 individual has genotype $MMQQNN$.

Table 8.1 displays marginal genotype probabilities for the F2, where the loci under consideration are M , Q and N . By definition, we have the relationships given in Equations (8.1) and (8.2) and the conditional genotype probabilities in Table 8.2 follow naturally.

$$p_{Q1} = P(\text{F1 transmits } Q \mid \text{F1 transmits } MN) = \frac{\pi_{00}}{1 - r_{MN}} = 1 - \frac{\pi_{11}}{1 - r_{MN}}. \quad (8.1)$$

$$p_{Q2} = P(\text{F1 transmits } Q \mid \text{F1 transmits } Mn) = \frac{\pi_{01}}{r_{MN}} = 1 - \frac{\pi_{10}}{r_{MN}}. \quad (8.2)$$

Therefore, from Table 8.2, we have simple expressions for the conditional probability $P(x_Q|x_M, x_N)$. Analogous expressions for $P(x_L|x_K, x_M)$ and $P(x_R|x_N, x_O)$ are easy to derive. By assuming independent crossovers, we calculate w_{ik} as given in Equation (5.20). This completes the specification of \mathbf{W} for the F2 design.

Table 8.1: Marginal genotype probabilities in an F2 population for a QTL (Q) and two flanking markers (M and N). The recombination probabilities π_{11} , π_{10} , π_{01} and π_{00} are defined in equations (2.1) to (2.4), with $A = M$ and $B = N$, and r_{MN} is the recombination fraction between M and N .

x_M, x_N	$P(x_M, x_N)$	$P(QQ, x_M, x_N)$	$P(Qq, x_M, x_N)$	$P(qq, x_M, x_N)$
$MMNN$	$0.25(1 - r_{MN})^2$	$0.25\pi_{00}^2$	$0.5\pi_{00}\pi_{11}$	$0.25\pi_{11}^2$
$MMNn$	$0.5r_{MN}(1 - r_{MN})$	$0.5\pi_{00}\pi_{01}$	$0.5(\pi_{00}\pi_{10} + \pi_{01}\pi_{11})$	$0.5\pi_{10}\pi_{11}$
$MMnn$	$0.25r_{MN}^2$	$0.25\pi_{01}^2$	$0.5\pi_{01}\pi_{10}$	$0.25\pi_{10}^2$
$MmNN$	$0.5r_{MN}(1 - r_{MN})$	$0.5\pi_{00}\pi_{10}$	$0.5(\pi_{00}\pi_{01} + \pi_{10}\pi_{11})$	$0.5\pi_{01}\pi_{11}$
$MmNn$	$0.5(1 - r_{MN})^2 + 0.5r_{MN}^2$	$0.5\pi_{00}\pi_{11} + 0.5\pi_{10}\pi_{01}$	$0.5(\pi_{00}^2 + \pi_{01}^2) + 0.5(\pi_{11}^2 + \pi_{10}^2)$	$0.5\pi_{10}\pi_{01} + 0.5\pi_{00}\pi_{11}$
$Mmnn$	$0.5r_{MN}(1 - r_{MN})$	$0.5\pi_{01}\pi_{11}$	$0.5(\pi_{00}\pi_{01} + \pi_{11}\pi_{10})$	$0.5\pi_{00}\pi_{10}$
$mmNN$	$0.25r_{MN}^2$	$0.25\pi_{10}^2$	$0.5\pi_{01}\pi_{10}$	$0.25\pi_{01}^2$
$mmNn$	$0.5r(1 - r_{MN})$	$0.5\pi_{10}\pi_{11}$	$0.5(\pi_{00}\pi_{10} + \pi_{01}\pi_{11})$	$0.5\pi_{00}\pi_{01}$
$mmnn$	$0.25(1 - r_{MN})^2$	$0.25\pi_{11}^2$	$0.5\pi_{00}\pi_{11}$	$0.25\pi_{00}^2$

Table 8.2: Conditional QTL genotypic probabilities in the F2 in terms of the conditional gene-transmission probabilities p_{Q1} and p_{Q2} .

x_M, x_N	$P(x_Q = QQ x_M, x_N)$	$P(x_Q = Qq x_M, x_N)$	$P(x_Q = qq x_M, x_N)$
$MMNN$	p_{Q1}^2	$2p_{Q1}(1 - p_{Q1})$	$(1 - p_{Q1})^2$
$MMNn$	$p_{Q1}p_{Q2}$	$p_{Q2}(1 - p_{Q1}) + p_{Q1}(1 - p_{Q2})$	$(1 - p_{Q1})(1 - p_{Q2})$
$MMnn$	p_{Q2}^2	$2p_{Q2}(1 - p_{Q2})$	$(1 - p_{Q2})^2$
$MmNN$	$p_{Q1}(1 - p_{Q2})$	$1 - p_{Q1} - p_{Q2} + 2p_{Q1}p_{Q2}$	$p_{Q2}(1 - p_{Q1})$
$MmNn$	$\frac{(1 - r_{MN})^2 p_{Q1}(1 - p_{Q1})}{(1 - r_{MN})^2 + r_{MN}^2} + \frac{r_{MN}^2 p_{Q2}(1 - p_{Q2})}{(1 - r_{MN})^2 + r_{MN}^2}$	$\frac{(1 - r_{MN})^2(1 - 2p_{Q1}(1 - p_{Q1}))}{(1 - r_{MN})^2 + r_{MN}^2} + \frac{r_{MN}^2(1 - 2p_{Q2}(1 - p_{Q2}))}{(1 - r_{MN})^2 + r_{MN}^2}$	$\frac{(1 - r_{MN})^2 p_{Q1}(1 - p_{Q1})}{(1 - r_{MN})^2 + r_{MN}^2} + \frac{r_{MN}^2 p_{Q2}(1 - p_{Q2})}{(1 - r_{MN})^2 + r_{MN}^2}$
$Mmnn$	$p_{Q2}(1 - p_{Q1})$	$1 - p_{Q1} - p_{Q2} + 2p_{Q1}p_{Q2}$	$p_{Q1}(1 - p_{Q2})$
$mmNN$	p_{Q2}^2	$2p_{Q2}(1 - p_{Q2})$	$(1 - p_{Q2})^2$
$mmNn$	$p_{Q1}p_{Q2}$	$p_{Q2}(1 - p_{Q1}) + p_{Q1}(1 - p_{Q2})$	$(1 - p_{Q1})(1 - p_{Q2})$
$mmnn$	p_{Q1}^2	$2p_{Q1}(1 - p_{Q1})$	$(1 - p_{Q1})^2$

The RIM1 model fits three QTL. Consequently, the F2 model has are $t = 27$ possible QTL genotypes. The matrix of category identities \mathbf{Z} is $n \times 27$, where n is the number of observed individuals. Likewise, the contrast matrix \mathbf{C} is $27 \times t'$ where t' is the number of contrasts being fitted. Now, we will briefly look at choosing QTL contrasts for use with the F2 model.

In the F2 there are three possible genotypes at each locus. Therefore, at most two contrasts can be fitted to extract the main effects at each locus. The QTL contrast matrix, \mathbf{C} , codes the intercept, main QTL effects and any interactions between QTL that we choose to fit. Suppose that we wish to fit only the main effects, then, for the RIM1 model, \mathbf{C} will have 27 rows and seven columns. Refer to locus L , Q and R , as the first, second and third QTL locus, respectively. To fit only the main effects, let $\mathbf{C}_{k1} = 1$ and for $p = 1, 2, 3$, define

$$\mathbf{C}_{k(2p)} = \begin{cases} 1, & \text{if QTL genotype } k \text{ is homozygous-high at the } p^{\text{th}} \text{ QTL locus} \\ 0, & \text{if QTL genotype } k \text{ is heterozygous at the } p^{\text{th}} \text{ QTL locus} \\ -1, & \text{if QTL genotype } k \text{ is homozygous-low at the } p^{\text{th}} \text{ QTL locus.} \end{cases}$$

$$\mathbf{C}_{k(2p+1)} = \begin{cases} 1, & \text{if QTL genotype } k \text{ is homozygous-high at the } p^{\text{th}} \text{ QTL locus} \\ -1, & \text{if QTL genotype } k \text{ is heterozygous at the } p^{\text{th}} \text{ QTL locus} \\ 1, & \text{if QTL genotype } k \text{ is homozygous-low at the } p^{\text{th}} \text{ QTL locus.} \end{cases}$$

Therefore, $\mathbf{C}_{\bullet 1} = \mathbf{1}_{27}$ and for $p = 1, 2, 3$, the column vector $\mathbf{C}_{\bullet(2p)}$ is a vector of contrast coefficients for extracting the additive effect of the homozygous-high at locus p , while $\mathbf{C}_{\bullet(2p+1)}$ is a vector of contrast coefficients for extracting its dominance effect. The parameter vector \mathbf{b} is associated with the columns of \mathbf{C} , where

$$\mathbf{b} = (b_0, b_1, b_2, b_3, b_4, b_5, b_6) = (b_0, a_L, d_{LL}, a_Q, d_{QQ}, a_R, d_{RR}),$$

with $a_L = \frac{1}{2}a_{LL}$, $a_Q = \frac{1}{2}a_{QQ}$ and $a_R = \frac{1}{2}a_{RR}$.

As with the backcross model, the matrix \mathbf{X}_2 codes genotypes at selected flanking markers, which are included to control the genetic background. As before, a parameter vector \mathbf{b}^* is associated with its columns. The matrix \mathbf{X}_2 may also contain non-genetic factors. This completes the RIM1 model-specification for the F2 breeding design.

8.2.2 Designs involving loci with more than two alleles

The RIM1 model readily extends to situations where more than two alleles may be present at any locus in the data. It places no restriction on the number of alleles at each locus because it is not defined in terms of allele counts. Rather, it is defined in terms of genotype categories and genotype probabilities (which are functions of recombination probabilities).

It is the structure of the breeding design and the number of distinct genotypes which determine the dimensionality and content of each matrix: \mathbf{Z} , \mathbf{C} , $\boldsymbol{\beta} = (\mathbf{b}, \mathbf{b}^*)^T$ and \mathbf{W} . The number of rows of \mathbf{C} is determined by the number of QTL genotypes. The number of columns of \mathbf{C} and the number of elements in \mathbf{b} is determined by the number of QTL effects that we want to fit. The number of rows of \mathbf{W} is determined by the number of distinct marker genotypes and the number of columns of \mathbf{W} by the number of distinct QTL genotypes. Once the structures of these matrices have been determined, the practical aspect of programming this extension to RIM1 may be addressed using the modular strategy described in last paragraph of Section 5.4.4.

The formulae for calculation the MLEs of $\boldsymbol{\beta}$ and σ will still be as given in Equations 5.67 and 5.68, respectively. However, the new structure of \mathbf{W} will require new formulae for the MLEs of the mixing parameters because, as described on page 101 below Equation 5.69, any formula for calculation $\hat{\phi}$ will depend on both the breeding design and the genetic mapping function being used.

When we consider implementing an extension involving more than two alleles at each locus, we must also consider what impact this will have on our information

matrix calculations. The overall form of the Fisher information matrix, as given in Equation (5.97), will not change. However, more effort may be needed to implement the formula because the last block, $\mathcal{I}_{\phi\phi}$, of the information matrix requires calculating a hessian involving the mixing parameters (see Equations 5.97 and 5.36). Like \mathbf{W} , this hessian will depend on both the breeding design and the genetic mapping function, and its complexity will depend on the complexity of \mathbf{W} .

We must also be aware that with more than two alleles at each locus, the numbers of marker and QTL genotypes involved may be very large indeed and so we need to make sure that the sample size is large enough to cater for the increase in the number of parameters to be estimated.

8.3 Applications to real data

In the previous chapter, the RIM1 model was applied to simulated backcross data. The results showed that the methodology proposed in Chapter 5 lead to improved tests for QTL. It is also important to assess the behaviour of RIM1 when applied to real data. Therefore the RIM1 model was also applied to two, publicly available, real datasets:

- The male F2 mouse dataset of Horvat and Medrano (1995), which is distributed with QTL Cartographer.
- The backcross drosophila dataset, BM2, of Zeng *et al.* (2000), obtained from the URL <ftp://statgen.ncsu.edu/pub/qtldcart/data/zengetal99/>.

There are several advantages to using public datasets to assess our new methodology. The main advantage is that it allows us to compare results from our new methodology with published QTL mapping results for the same datasets. This provides a rough benchmark for assessing the performance of the new method.

8.3.1 Real F2 Application

The aim of Horvat and Medrano (1995) was to locate the ‘high growth’ locus, a region in the mouse genome that increases both weight gain and body size in mature mice. Analysis was restricted to chromosome 10 because of prior research. They developed a mouse dataset based on 190 male individuals from an F2 population. The trait was weight gain (in grams) from 14 to 63 days of age.

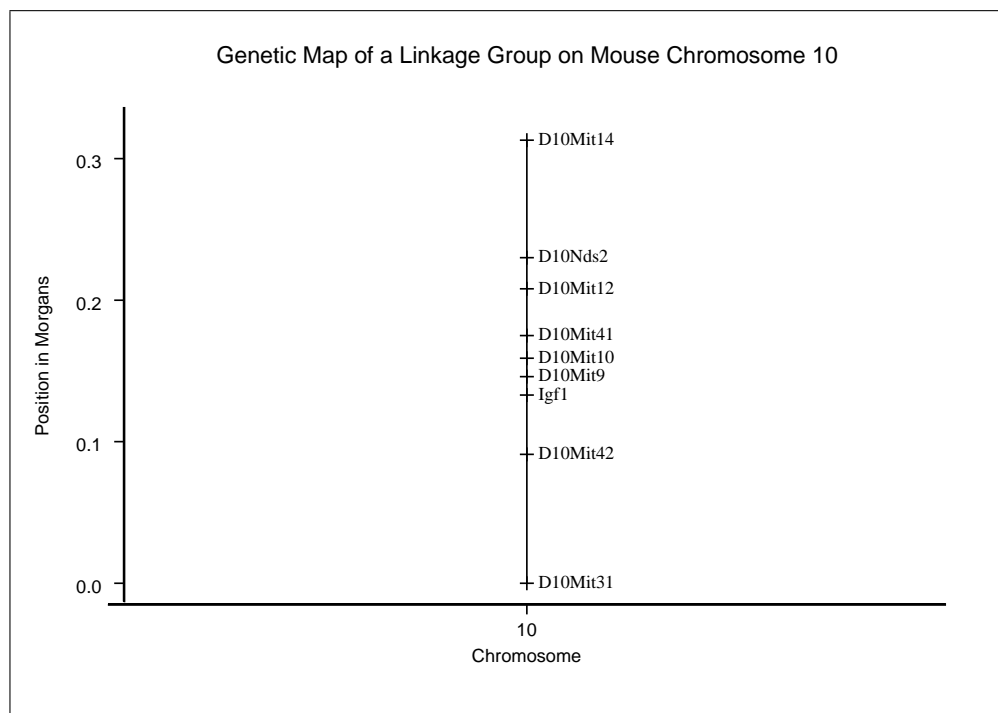


Figure 8.1: Horvat and Medrano mouse map.

The genotypes of the mice were recorded for nine markers on chromosome 10. The nine markers are *D10Mit31*, *D10Mit42*, *Igf1*, *D10Mit9*, *D10Mit10*, *D10Mit41*, *D10Mit12*, *D10Nds2* and *D10Mit14*. Figure 8.1 shows the corresponding marker genetic map.

In these F2 males, Horvat and Medrano found a high growth (*hg*) QTL between *D10mit* and *D10mit12*, located at distance of 1.5 centi-Morgans distal to *D10mit41*.

In their QTL detection methodology, they took a logarithmic transformation of the trait values. Then they used the MAPMAKER/QTL 1.1 software (of Lincoln *et al.*, 1992) to perform Lander-Botstein interval mapping based on the transformed trait. Their hypothesis tests were based on the LOD ratio statistic, which differs from the likelihood ratio test statistic by a constant. They used the permutation method of Churchill and Doerge (1994) to obtain an empirical estimate of the null distribution of the LOD statistic. From this distribution, they determined empirical threshold values for QTL detection.

Lander-Botstein Interval mapping (also called simple interval mapping) does not fit marker cofactors and it does not assume a mixture distribution for the trait. Rather, it assumes a single Normal distribution. As Horvat and Medrano used simple interval mapping, it was necessary for them to transform the trait values to remove or reduce non-normality. In contrast, our new method, RIM1 assumes a mixture distribution for the trait values. Therefore, for this dataset, it was neither necessary nor desirable for us to make a normalizing transform (or change of scale). Instead, we directly analysed the trait values (weight gain, in grams, from 14 to 63 days of age). Details of configuring the data for analysis with RIM1 are given in Appendix B.7.

The availability of the Fisher information matrix within the RIM1 procedure is advantageous because it dramatically reduces the computational burden of interval mapping. For example, consider the permutation tests carried out by Horvat and Medrano on this mouse dataset. There are eight marker intervals in this dataset. In each interval, calculation of the LOD ratio statistic (LOD score) requires two maximum likelihood calculations. This is because the likelihood must be calculated under both the null and alternative hypotheses. Each maximum likelihood calculation involves an iterative computation.

To analyse the eight marker intervals, Horvat and Medrano, carried out 1000

permutations. This required 1616 maximizations of the likelihood function (16 for the original dataset and 1600 for the re-samples). In contrast, the RIM1 method required only eight maximizations, one for each testing interval. Note that these tests are based directly on the MLEs and their standard errors, and that the standard errors are obtained by direct calculation of the Fisher information matrix using the formula given in Equation (5.97).

Table 8.3 displays the results of applying RIM1 to the Horvat and Medrano mouse data. RIM1 detected a *hg* QTL between D10MIT41 and D10MIT12 at significance level 0.01 (p-value for $T_1(\hat{a}_Q) = 0.0013$, p-value for $J_c = 0.0007$). The MLE for the recombination fraction between the QTL and marker $M = D10Mit41$ was $r_{MQ} = 0.01496$. This corresponds genetic distance of 1.51 centi-Morgans distal to *D10Mit41*. A 99% confidence interval for the distance ($\widehat{\text{dist}}(MQ)$) between the *D10Mit41* locus and the *hg* QTL is (0.3, 2.7), which is 2.4 centi-Morgans wide. Note that, with less computational effort, the RIM1 procedure lead to the same MLE as that obtained by Horvat and Medrano (1.5 cM to one decimal place).

Table 8.3: Results of applying RIM1 to the F2 Mouse dataset of Horvat and Medrano (1995). Shown: estimated additive (\hat{a}_Q) and dominance (\hat{d}_{QQ}) effects, together with standard errors (SD); p-values of test statistics $T_1(\hat{a}_Q)$, $T_1(\hat{d}_{QQ})$, for non-zero effect; p-value of test J_c for whether QTL is interior to the interval; the estimated error variance ($\hat{\sigma}^2$); inter-locus distances $\text{dist}(MN)$, $\widehat{\text{dist}}(MQ)$ in centi-Morgans. Asterisks mark significant p-values.

Interval	MLE	SD	MLE	SD	P-value	P-value	MLE	Actual	MLE	P-value
	\hat{a}_Q	\hat{a}_Q	\hat{d}_{QQ}	\hat{d}_{QQ}	$T_1(\hat{a}_Q)$	$T_1(\hat{d}_{QQ})$	$\hat{\sigma}^2$	MN	$\widehat{\text{dist}}(MQ)$	$J_c(\hat{p}_{Q2})$
<i>D10Mit31</i> – <i>D10Mit42</i>	–1.17	0.55	–0.13	0.31	0.033*	0.682	8.36	9.1	0.0	0.493
<i>D10Mit42</i> – <i>Igf1</i>	1.81	0.79	0.97	0.45	0.021*	0.029	8.36	4.2	0.0	0.495
<i>Igf1</i> – <i>D10Mit9</i>	1.15	1.51	–1.10	0.78	0.446	0.158	8.36	1.3	0.0	
<i>D10Mit9</i> – <i>D10Mit10</i>	–3.29	1.74	–0.51	0.88	0.058	0.560	8.36	1.3	0.7	
<i>D10Mit10</i> – <i>D10Mit41</i>	2.24	1.35	1.17	0.71	0.097	0.100	8.29	1.6	1.6	
<i>D10Mit41</i> – <i>D10Mit12</i>	4.78	1.50	1.39	0.75	0.001**	0.063	7.71	3.3	1.5	0.001***
<i>D10Mit12</i> – <i>D10Nds2</i>	0.05	0.91	–0.14	0.44	0.952	0.758	7.70	2.2	2.2	
<i>D10Nds2</i> – <i>D10Mit14</i>	–1.32	44.89	0.38	22.45	0.977	0.987	8.16	8.3	8.3	

Doerge *et al.* (1997) also analysed the data of Horvat and Medrano. Assuming that the quoted threshold value of 9.68 relates to $H_1:H_3$, the results in their Table 4, agree with the results shown in Table 8.3 of this thesis: (a) a single QTL found; (b) in the same interval $D10mit41 - D10mit12$; (c) in the same place (with Doerge *et al.* giving only the grid point closest to the position estimated by RIM1). The hypothesis from Doerge *et al.* Table 4 that is being tested in Table 8.3, is $H_0 : a = 0$ and $d = 0$ versus $H_4 : \text{at least one of } a \text{ and } d \text{ is non zero}$. That is, H_0 versus (H_1 or H_2 or H_3).

Despite the relatively small sample of 190 individuals, this QTL was easy to detect because it is an isolated QTL with fairly large effects. The estimated additive allelic effect was $a_Q = 4.78$ grams which implies that the additive genotypic effect is $a_{QQ} = 9.56$ grams.

8.3.2 Real Backcross Application

Zeng *et al.* (2000) used QTL mapping to explore the genetic basis for observed differences, in a morphological character, between males from two closely related *Drosophila* species: *Drosophila mauritiana* and *Drosophila simulans*. The morphological trait studied was the size and shape of the posterior lobe of the male genital arch. This trait was quantified as the average over both sides of the first principal component of the Fourier coefficients (see Kuhl and Giardina, 1982) of the posterior lobe. The resulting trait values, denoted PC1, were used together with marker data in order to map QTL controlling this morphological character. The PC1 trait assay methodology is described in Liu *et al.* (1996), and the techniques used for genetic-marker data acquisition is described in Zeng *et al.* (2000).

Four backcross samples were created and analysed by Zeng *et al.* (2000):

BM1 Backcross: $F1 \times D. mauritiana$, 192 individuals;

BM2 Backcross: $F1 \times D. mauritiana$, 299 individuals;

BS1 Backcross: $F1 \times D. simulans$, 186 individuals;

BS2 Backcross: $F1 \times D. simulans$, 288 individuals.

(Note: $F1$ is $D. mauritiana \times D. simulans$.)

We do not aim to re-analyse all of these samples. Rather, we aim to illustrate the results of applying RIM1 to any real backcross sample. Therefore, in this section, we illustrate the behaviour of RIM1 by applying it to only one of these samples: BM2.

The linkage map of markers for sample BM2 contained 42 loci having the names and locations listed below.

1. The first six markers are located on chromosome X and are named: *ewg*, *w*, *RpS6*, *v*, *Sd*, *run*. The distances (in cM) between adjacent markers in this linkage group are: 3.60, 10.60, 9.20, 17.20, 18.70.
2. The next 13 markers are located on chromosome 2 and are named: *gl*, *Pgk*, *Cg25C*, *Gpdh*, *ninaC*, *Glt*, *Mhc*, *DoxA2*, *DucC*, *sli*, *Egfr*, *twi*, *zip*. The distances (in cM) between adjacent markers in this linkage group are: 6.98, 10.10, 4.94, 6.51, 6.19, 33.24, 3.90, 4.55, 42.06, 37.51, 21.19, 3.71, 7.03.
3. The next 22 markers are located on chromosome 3 and are named: *Lsp1*, *ve*, *Acr64B*, *Dbi*, *h*, *CycA*, *fz*, *Eip71CD*, *tra*, *rdgC*, *5-HT2*, *Antp*, *ninaE*, *Fas1*, *Mst*, *Odh*, *Tub85E*, *hb*, *Rox8*, *Ald*, *Mlc1*, *jan*, *Ef1d2*. The distances (in cM) between adjacent markers in this linkage group are: 4.99, 9.34, 6.97, 7.44, 14.46, 6.79, 3.55, 6.32, 11.86, 4.58, 6.85, 6.35, 11.79, 12.88, 9.15, 3.30, 7.98, 13.09, 10.04, 3.70, 9.79, 3.43.

Of the 299 cases in sample BM2, four cases had a missing value for the trait PC1, and 89 cases had a missing genotype at least one marker locus. The paper by Zeng *et al.* (2000) does not document their strategy for handling cases with missing marker and/or trait data.

There are usually two options when working with missing data: either omit cases or replace the missing data with imputed data. In QTL mapping, we are trying to detect effects which may be very weak and subtle, and putting imputed trait data into the method is too great a risk. Therefore, it is best to omit cases with missing *trait* data.

Where *marker* data are missing, we may either replace them with imputed data or we may omit the corresponding record(s) from the analysis. One method for imputing missing marker data for an individual is to replace the missing data with their conditional expectations given all the observed marker genotypes for that individual (Jiang and Zeng, 1997). Another method is to randomly assign a genotype by sampling from the conditional distribution of the individual's missing marker genotype given his/her observed trait and marker data (Yi *et al.*, 2003). This conditional distribution may be estimated from the E-step in an implementation of the EM algorithm.

When omitting cases with missing marker data, one option is to throw away all such cases. Another option is to include those cases when considering intervals on chromosomes where the marker data are complete (or at the very least, only when testing several intervals away from where the marker data are missing).

For simplicity, all records having missing marker and/or trait data were removed from the BM2 sample and the RIM1 model was implemented using the remaining 210 records. All available background markers were included as cofactors in the model. However, no interactions between QTL were modelled. RIM1 is a multi-QTL model because it fits three QTL, one in a central testing interval and two background QTL, one on each side of the testing interval. The three linkage groups were scanned for

QTL by sliding the testing interval along each linkage group, and re-fitting RIM1 for each testing interval. Table 8.4 shows the results of applying RIM1 to the BM2 backcross sample. In Table 8.4, the column entitled ‘Zeng $\widehat{\text{dist}}(Q)$ ’ shows the locations of QTL found by Zeng *et al.* (2000) using a method called Multiple Interval Mapping.

Multiple Interval Mapping (MIM) is a stepwise selection model proposed by Kao *et al.* (1999) as an extension to Composite Interval Mapping. MIM starts with a set of QTL at locations determined by prior CIM modelling, and builds a final model through several rounds of forward/backward selection. Hypothesis testing with MIM is based on the LOD score statistic with critical values calculated either from the traditional LOD cut-off point of 4.4, or from permutation tests. Using the BM2 sample, and fitting MIM with epistatic interactions between QTL, Zeng *et al.* found 15 QTL. It is useful to compare our RIM1 results with those from MIM because both models fit multiple QTL.

Applying RIM1 to BM2 (without any interaction terms) revealed 18 QTLs having significant effects at the 5% significance level. However, only 12 of these effects were significant at the 0.1% significance level (see Table 8.4). Eleven of the QTL detected by RIM1 were in similar locations to those found by Zeng *et al.*. The QTL effects \widehat{b}_Q displayed in Table 8.4 are opposite in sign to those presented by Zeng *et al.* because the genotypes were coded differently. In RIM1, the homozygous *mauritiana* genotype was coded as $QQ = 1$ and the heterozygous F1 genotype as $Qq = 0$.

Although 18 significant QTL effects were found, there was not sufficient evidence that they were all interior to their respective testing intervals. In the joint test for QTL effect and location, only seven (7) QTL were found to be interior to the corresponding testing interval and only two (2) of these were significant at the 0.1% significance level. These results suggest that while a sample of size 210 was useful for detecting QTL, the sample size was too small to precisely determine QTL location.

Table 8.4: RIM1 results for the *Drosophila* dataset BM2 of Zeng *et al.* (2000). Cases with missing marker/trait data removed ($n = 210$). The MLE \hat{b}_Q and its standard error (SD) are in PC1 units; p-value of test $T_1(\hat{b}_Q)$ for non-zero effect; p-value of test $J_c(\hat{p}_{Q2})$ for whether QTL is interior to interval; asterisks mark significant p-values; map distances $\text{dist}(M)$, $\widehat{\text{dist}}(Q)$ in centi-Morgans. The column Zeng $\widehat{\text{dist}}(Q)$ shows the locations of QTL found by Zeng *et al.*

Interval $M - N$	MLE $\hat{b}_Q \times 10^3$	SD $\times 10^3$ of \hat{b}_Q	P-value $T_1(\hat{b}_Q)$	Actual $\text{dist}(M)$	Zeng $\widehat{\text{dist}}(Q)$	MLE $\widehat{\text{dist}}(Q)$	P-value $J_c(\hat{p}_{Q2})$	99.9% CI for MLE $\widehat{\text{dist}}(Q)$
<i>ewg - w</i>	-1.67	0.47	0.000 ***	0.00	1	0.00	0.498	[0.00, 0.02]
<i>w - RpS6</i>	1.24	1.18	0.295	3.60		3.60		
<i>RpS6 - v</i>	-2.32	0.46	0.000 ***	14.20	20	18.51	0.002 **	[16.17, 20.53]
<i>v - Sd</i>	-0.14	0.43	0.742	23.40		32.51		
<i>Sd - run</i>	0.85	0.28	0.003 **	40.60		47.53	0.491	[47.51, 47.53]
<i>gl - Pgk</i>	-1.50	0.41	0.000 ***	0.00		2.87	0.002 **	[0.00, 6.08]
<i>Pgk - Cg25C</i>	-0.16	0.57	0.774	6.98	10	14.93		
<i>Cg25C - Gpdh</i>	0.01	0.55	0.981	17.08		17.08		
<i>Gpdh - ninaC</i>	-2.19	0.55	0.000 ***	22.02	26	24.52	0.001 ***	[23.02, 25.95]
<i>ninaC - Glt</i>	0.14	0.45	0.758	28.53		31.82		
<i>Glt - Mhc</i>	-0.92	0.87	0.286	34.72		46.85		
<i>Mhc - Dox2</i>	-0.97	0.54	0.070	67.96	69	68.04		
<i>Dox2 - DucC</i>	-0.67	0.54	0.211	71.86		72.89		
<i>DucC - sli</i>	-1.95	0.38	0.000 ***	76.41		82.13	0.490	[82.13, 82.13]
<i>sli - Egfr</i>	-1.95	0.38	0.000 ***	113.92	114	113.92	0.458	[113.92, 113.94]
<i>Egfr - twi</i>	2.25	0.49	0.000 ***	135.11	135	135.11	0.496	[135.11, 135.11]
<i>twi - zip</i>	-2.70	0.39	0.000 ***	138.82	143	139.15	0.013 *	[139.03, 139.23]
<i>Lsp1 - ve</i>	-0.31	0.39	0.414	0.00		0.00		
<i>ve - Acr64B</i>	-0.26	0.65	0.684	4.99	5	4.99		
<i>Acr64B - Dbi</i>	-2.15	0.51	0.000 ***	14.33	17	15.61	0.004 **	[14.33, 17.18]
<i>Dbi - h</i>	-1.62	0.44	0.000 ***	21.30		24.15	0.001 **	[22.36, 25.81]
<i>h - CycA</i>	-1.20	0.54	0.026 *	28.74		28.75	0.430	[28.74, 28.87]
<i>CycA - fz</i>	-2.70	0.59	0.000 ***	43.20	47	44.41	0.000 ***	[43.24, 45.58]
<i>fz - Eip71CD</i>	0.08	0.70	0.910	49.99		51.25		
<i>Eip71CD - tra</i>	0.11	0.53	0.833	53.54		55.57		
<i>tra - rdgC</i>	0.32	0.64	0.617	59.86		63.05		
<i>rdgC - 5-HT2</i>	0.40	0.59	0.496	71.72		71.72		
<i>5-HT2 - Antp</i>	-3.48	0.75	0.000 ***	76.30	83	77.69	0.497	[77.69, 77.69]
<i>Antp - ninaE</i>	-0.12	0.68	0.863	83.15		84.28		
<i>ninaE - Fas1</i>	-0.44	0.44	0.320	89.50		91.25		
<i>Fas1 - Mst</i>	-0.43	0.48	0.369	101.29		101.29		
<i>Mst - Odh</i>	-0.50	0.48	0.303	114.17	117	114.17		
<i>Odhd - Tub85E</i>	-1.49	0.70	0.034 *	123.32		123.59	0.495	[123.59, 123.59]
<i>Tub85E - hb</i>	0.11	0.71	0.873	126.62		127.21		
<i>hb - Rox8</i>	-1.17	0.50	0.019 *	134.60	141	135.38	0.493	[135.38, 135.38]
<i>Rox8 - Ald</i>	0.91	0.82	0.267	147.69		148.16		
<i>Ald - Mlc1</i>	0.89	0.58	0.127	157.73		157.73		
<i>Mlc1 - jan</i>	-1.52	0.75	0.043 *	161.43	168	161.46	0.076	[161.43, 161.53]
<i>jan - Efd2</i>	-1.30	0.45	0.004 **	171.22		171.22	0.474	[171.22, 171.22]

In Chapter 7, simulated backcross data revealed that, for small sample sizes, the information matrix tends to underestimate the standard errors. Also, in intervals without QTL, lack of identifiability of p_{Q2} breaks down the ability of the information matrix to correctly estimate the standard errors of p_{Q2} . This latter situation does not present a problem because we are only interested in QTL location if the corresponding QTL effect is significant. To examine how the information matrix behaved for this real backcross sample, 1000 bootstrap samples were created using simple random sampling (from BM2) with replacement. As with the simulated data, the information matrix also underestimated standard errors for this real sample (see Table 8.5).

Table 8.5: Results of bootstrapping the *Drosophila* dataset BM2 of Zeng *et al.* (2000). Looking at chromosome 2 only. The MLEs and the asymptotic standard errors (imat SD) are from the original sample. The bootstrap standard errors (boot SD) are based on 1000 bootstrap replicates.

Interval $M - N$	MLE $\hat{b}_Q \times 10^3$	imat SD $\times 10^3$ of \hat{b}_Q	boot SD $\times 10^3$ of \hat{b}_Q	MLE \hat{p}_{Q2}	imat SD of \hat{p}_{Q2}	boot SD of \hat{p}_{Q2}
<i>gl - Pgk</i>	-1.50 ***	0.41	0.87	0.56 **	0.15	0.40
<i>Pgk - Cg25C</i>	-0.16	0.57	1.06	~ 0.00	~ 0.00	0.48
<i>Cg25C - Gpdh</i>	0.01	0.55	1.23	~ 1.00	~ 0.00	0.46
<i>Gpdh - ninaC</i>	-2.19 ***	0.55	1.60	0.36 ***	0.12	0.38
<i>ninaC - Glt</i>	0.14	0.45	1.18	~ 0.00	~ 0.00	0.42
<i>Glt - Mhc</i>	-0.92	0.87	1.63	~ 0.00	~ 0.00	0.49
<i>Mhc - DoxA2</i>	-0.97	0.54	1.99	0.92	0.12	0.39
<i>DoxA2 - DucC</i>	-0.67	0.54	1.76	~ 0.00	~ 0.00	0.39
<i>DucC - sli</i>	-1.95 ***	0.38	0.86	~ 0.00	~ 0.00	0.32
<i>sli - Egfr</i>	-1.95 ***	0.38	1.50	~ 1.00	~ 0.00	0.26
<i>Egfr - twi</i>	2.25 ***	0.49	1.05	~ 1.00	~ 0.00	0.33
<i>twi - zip</i>	-2.70 ***	0.39	1.05	0.20 *	0.09	0.24

8.4 Overview

First, this chapter outlined strategies for fitting interactions between QTL as part of the RIM1 model, and for applying RIM1 to the F2 and other breeding designs. Then, RIM1 was applied to two real datasets. In both of these datasets, RIM1 successfully detected QTL, and the results of RIM1 agreed well with QTL mapping results published by other researchers.

Chapter 9

Summary and Conclusions

This thesis explored the mixture model, developed a new extension to Composite Interval Mapping and derived new matrix formulae which makes the evaluation of the Fisher information matrix tractable for Normal mixtures having an arbitrary number of mixing components.

A new extension to Composite Interval Mapping was devised. The new model, RIM1, simultaneously conditions on four markers to increase the precision of interval mapping in the presence of multiple QTL. RIM1 fits exactly three putative QTL, one in each of three contiguous intervals. Applications to simulated and real data showed that RIM1 had strong power to detect QTL while dramatically decreasing the rate of the of false detections. For large samples, RIM1, was shown to dramatically reduce ghosting (when compared with CIM) while retaining high power to detect QTL.

One might ask whether the robustness against ghosting, exhibited by RIM1, is due to the modelling itself, or to the choice of estimation procedure. The answer is that the robustness is due to the modelling itself. In particular, it results from fitting flanking QTL in the RIM1 model. This is supported by the fact that the same estimation procedure was used for CIM and RIM1. The commonality in the estimation procedure for these models is explained in Section 5.4.4. With RIM1,

similar results are obtained when using the joint hypothesis test for QTL effect and position or the test for QTL effects only. This also suggests that the robustness of RIM1 against ghosting is a result of fitting the flanking QTL. It is also interesting to note that RIM1 is more susceptible than CIM to the problems of a multi-modal likelihood function. Despite this disadvantage, the structure of the RIM1 model enabled better control of ghosting than CIM.

Rather than working directly with the recombination fractions, the mixing proportions were expressed in terms of the conditional genotype transmission-probabilities (for example p_{Q1} and p_{Q2}). This allows flexibility, because one does not need to assume any specific three-locus mapping function within intervals. Also the resulting expressions have a simple form, which provides freedom from the need for the common, simplifying assumption that there are no double crossovers within intervals.

The problem of estimating the standard errors of parameters in a mixture model was addressed through direct calculation of the Fisher Information matrix. New matrix formulae were derived, allowing exact and convenient calculation of the Fisher information matrix in the context of Multinomial mixtures of Univariate Normal distributions. These matrix formulae hold for Normal mixture models with:

- any number of mixing components (for example, models with any number of QTL);
- any number of extra cofactors (but missing data is not allowed at the cofactors);
- any number of interactions between the missing factors (for example, interactions between different QTL),
- any number of interactions between extra cofactors (for example, fitting interactions between background markers may be useful for capturing interactions between background QTL that are not explicitly included in the model).

- different mixing proportions for distinct subgroups within a sample. (One only needs to specify the relevant mixing proportions. This allows conditioning on different marker groups. It also allows conditioning on different crosses, thereby facilitating simultaneous analysis of multiple crosses.)

In addition, the proposed formulae do not require element-wise evaluation of the Fisher information matrix. This makes the information matrix calculation practical to implement, irrespective of the number of mixing components involved. Program code in the R statistical language is provided in Appendix B as an illustration of how easily the matrix formulae can be programmed using a statistical package that allows matrix manipulation. The programs also illustrate how the the information matrix formulae readily accommodates different numbers of mixing parameters for different models.

This contribution is not only useful for QTL mapping problems. It is also useful for any statistical application that uses likelihood-based estimation with mixtures of univariate Normal distributions. Note, however, that these formulae do not hold for:

- mixture models with interactions between missing components and observed variables. Such mixtures become relevant, for example, if we wish to model QTL by environment interactions. This is one area for further development.
- models involving mixtures of Normals that have different variances (different values for σ^2).

The availability of the information matrix formulae allowed the development of improved hypothesis tests to reduce ghosting in both Composite Interval Mapping and RIM1. The information-matrix formulae provided here, are exact evaluations and so the requirements for them to be valid do not depend on any extra assumptions on top of those needed for the asymptotic maximum likelihood theory to hold.

If the sample size is large enough, the standard errors produced are expected to be similar to those obtainable from bootstraps and permutation-based re-sampling methods. Moreover, this method requires less computing time to compute a threshold (or critical value) for hypothesis testing than do permutation methods. There is also no need to compute the model under the null hypothesis. Simulations showed that the proposed method causes Composite Interval Mapping to be more robust against false detections, than do Likelihood Ratio Tests (LRT) based on a chi-square distribution with one degree of freedom.

The simulated results presented in this thesis show that small sample sizes can adversely affect the stability of the maximum likelihood estimates generated by both CIM and RIM1. Multi-QTL models such as RIM1 will suffer more greatly from lack of identifiability than CIM. Such models will perform better under large sample sizes. Note that by using the inverse Fisher information matrix as the variance-covariance matrix of the MLE's we are invoking asymptotic results. Therefore, it is not surprising that the results do not hold for small samples.

When the sample size is too small, the Fisher information matrix severely underestimates the standard errors of parameters in the mixture model. For example, the variance of \hat{b}_Q will tend to be underestimated for small samples. Unless the estimated effect \hat{b}_Q is itself close to zero, underestimating the variance of \hat{b}_Q can increase the risk of detecting ghost QTL effects when the statistic $T_1(b_Q)$ is used. Simultaneously, there will be an underestimate of the variance of \hat{p}_{Q2} . This will push the estimate of $J_c(p_{Q2})$ into the extreme tails of its distribution, so we will tend to accept the null hypothesis that the QTL is not interior to the testing interval. This indicates that, while it may be possible to detect QTL with small sample sizes, we generally will not be able to put much confidence in the estimated QTL locations. This behaviour is not totally undesirable because, in the presence of an unfavourably small sample, at least we will not place too much confidence in what could be spurious results.

For any specified model, statistical test and significance level, the power to detect QTL will be affected by the following factors (Liu, 1997, page 481):

- the number of QTLs affecting the trait and their genomic locations
- the linkage map density and coverage
- the distribution of QTL effects and the existence of gene interactions
- gene and genotype probabilities in the mapping population
- heritability of the trait
- sample size.

Further exploration of the RIM1 model could include an investigation of its behaviour as these factors change. An important and related consideration is how to select an appropriate sample size. This is often problematic because, for example, estimates of heritability and of the proportion of variation explained by QTL are not available *a priori*. QTL by environment interactions can also complicate the situation, and so this is an extension that merits further investigation.

The J_1 test was introduced as a method for controlling ghosting in CIM. However there are problems with the distribution of the J_1 test statistic. Therefore, to control ghosting, it is recommended that RIM1 be used instead of CIM, and that RIM1 should be used with the T_1 test statistic rather than the J_1 statistic.

Appendix A

Constructing an Orthogonal Contrast Matrix

We wish to find a matrix $\mathbf{C}_{k \times k-1}$ so that $[\mathbf{1}_n \quad \mathbf{X}\mathbf{C}]$ is orthogonal, where \mathbf{X} is an $n \times k$ binary matrix. One method is to use orthogonal polynomial coefficients. If the levels of our factor are evenly spaced then such coefficients have a convenient interpretation as coefficients of an orthogonal polynomial model of order $k-1$. Orthogonal polynomial coefficients are tabulated in the literature (see for example Draper and Smith (1998)).

Below, we propose two algorithms for obtaining a set of orthogonal contrasts.

Algorithm A.1. *Method 1 to find an orthogonal contrast matrix.*

1. Select a matrix $\check{\mathbf{C}}_{k \times k-1}$ such that $[\mathbf{1}_n \quad \mathbf{X}\check{\mathbf{C}}]$ has rank k .
2. Let $\mathbf{R} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k-1})$, where

$$\begin{aligned} \mathbf{v}_0 &= \mathbf{1}_n \\ \mathbf{v}_t &= \mathbf{X}\check{\mathbf{C}}_{\bullet t} - \sum_{h=0}^{t-1} \frac{\mathbf{v}_h^T \mathbf{X}\check{\mathbf{C}}_{\bullet t}}{\mathbf{v}_h^T \mathbf{v}_h} \mathbf{v}_h, \text{ for } t = 1, 2, \dots, k-1. \end{aligned}$$

3. $\mathbf{C} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{R}$ is an orthogonal contrast matrix. □

Proof that Algorithm A.1 produces an orthogonal contrast matrix. By construction, the matrix $[\mathbf{1}_n \quad \check{\mathbf{X}}\mathbf{C}]$ has linearly independent columns. We also assume here that the Columns of \mathbf{X} are linearly independent, so that the left inverse of \mathbf{X} exists. This is the standard assumption for regression on \mathbf{X} . Any matrix consisting of linearly independent columns can be transformed into an orthogonal matrix via the Gram-Schmidt Orthogonalisation process. Cadogan (1987), for example, provides a proof of this well established algebraic result. Step two applies the Gram-Schmidt process to $[\mathbf{1}_n \quad \check{\mathbf{X}}\mathbf{C}]$ to obtain an orthogonal matrix $\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{k-1}) = [\mathbf{1}_n \quad \mathbf{R}]$. We simply solve the equation $\mathbf{X}\mathbf{C} = \mathbf{R}$, by pre-multiplying both sides by the left inverse of \mathbf{X} to obtain $\mathbf{C} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}$. Then $[\mathbf{1}_n \quad \mathbf{X}\mathbf{C}] = \mathbf{V}$, which is orthogonal. Hence the algorithm produces an orthogonal contrast matrix \mathbf{C} . \square

Algorithm (A.1) is valid even if \mathbf{X} is not a binary incidence matrix. The above algorithm may be computationally intensive to implement if \mathbf{X} is large, so a more elegant algorithm for obtaining an orthogonal contrast matrix by using only the numbers of elements in each category is proposed below. However, the second method requires that \mathbf{X} is a binary incidence matrix.

Algorithm A.2. *Method 2 for obtaining an Orthogonal contrast matrix.*

1. Let

$$\mathbf{a} = (\sqrt{n_1}, \sqrt{n_2}, \dots, \sqrt{n_k})^T,$$

$$\mathbf{D} = \text{diag}(\sqrt{n_1}, \sqrt{n_2}, \dots, \sqrt{n_k}) \text{ and}$$

$$\mathbf{E} = [\mathbf{I}_{k-1} \quad \mathbf{0}_{k-1}]^T,$$

where $\mathbf{0}_{k-1}$ is a vector of $(k-1)$ zero elements and \mathbf{I}_{k-1} is the identity matrix of order $(k-1)$. The quantity n_i denotes the number of observations in category i (i.e. n_i is the number nonzero elements in the i^{th} column of the binary matrix \mathbf{X}) for $i = 1, \dots, k$.

2. Let $\mathbf{R} = (\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{k-1})$, where

$$\begin{aligned} \mathbf{v}_0 &= \mathbf{a} \\ \mathbf{v}_t &= \mathbf{D}\mathbf{E}_{\bullet t} - \sum_{h=0}^{t-1} \frac{\mathbf{v}_h^T \mathbf{D}\mathbf{E}_{\bullet t}}{\mathbf{v}_h^T \mathbf{v}_h} \mathbf{v}_h, \text{ for } t = 1, 2, \dots, k-1. \end{aligned}$$

3. $\mathbf{C} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{R}$ is an orthogonal contrast matrix. \square

Proof that Algorithm A.2 produces an orthogonal contrast matrix. By definition of its component matrices, it is clear that the matrix $[\mathbf{a} \ \mathbf{D}\mathbf{E}]$ has linearly independent columns. Step two in the algorithm applies the Gram-Schmidt process to the matrix $[\mathbf{a} \ \mathbf{D}\mathbf{E}]$ to obtain an orthogonal matrix $\mathbf{V} = (\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_{k-1}) = [\mathbf{a} \ \mathbf{R}]$. We simply solve the equation $\mathbf{D}\mathbf{C} = \mathbf{R}$, by pre-multiplying both sides by the left inverse of \mathbf{D} to obtain

$$\mathbf{C} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{R}.$$

Now, the matrix $[\mathbf{a} \ \mathbf{D}\mathbf{C}]$ is orthogonal by construction. Therefore $\mathbf{a}^T \mathbf{D}\mathbf{C}_{\bullet i} = 0$ and $\mathbf{C}_{\bullet i}^T \mathbf{D}^T \mathbf{D}\mathbf{C}_{\bullet j} = 0$ for $i \neq j$. However, $\mathbf{1}_n^T \mathbf{X}\mathbf{C}_{\bullet i} = \mathbf{a}^T \mathbf{D}\mathbf{C}_{\bullet i}$ and for $1 \leq i \leq k-1$. Also $\mathbf{C}_{\bullet i}^T \mathbf{X}^T \mathbf{X}\mathbf{C}_{\bullet j} = \mathbf{C}_{\bullet i}^T \mathbf{D}^T \mathbf{D}\mathbf{C}_{\bullet j}$ for $1 \leq i, j \leq k-1$. This implies that $[\mathbf{1}_n \ \mathbf{X}\mathbf{C}]$ is orthogonal, so is clear that \mathbf{C} is an orthogonal contrast matrix. \square

Appendix B

Programs and Code

The R language and environment (R Development Core Team (2006)) is well suited for our programming needs because its matrix and list objects offer flexible indexing and manipulation features. This appendix contains R code for implementing RIM1. It also contains examples of applying these programs to actual data analyses.

Section B.1 provides R code for parameter estimation in RIM1, CIM and six other models. Section B.2 provides a number of utility functions for QTL mapping and for importing QTL Cartographer input and output files. Section B.4 contains R code to implement the information matrix formulas. Usage examples are given in Sections B.3, B.5, B.6 and B.7. The code in Section B.6 implements a permutation method which randomises the covariate of interest among the sampled individuals. The example in B.7 illustrates the analysis of the F2 mouse data from Horvat and Medrano (1995).

B.1 R code for parameter estimation in RIM1 and its sub-models

Table B.1: List of functions used for model fitting

Function	Description	Dependencies
<code>rim.linecross()</code>	Main function for fitting RIM1 and sub-models. Calling this function will fit the models, calculate the information matrix and perform hypothesis testing. Its Return value includes the logarithm of the likelihood, maximum likelihood estimators (MLEs), standard errors of MLEs, optionally-the covariance matrix, p-values, and other informational output.	<code>validate.cross()</code> <code>cofactor.matrix()</code> <code>qtl.design()</code> <code>contrasts.b1()</code> <code>contrasts.f2()</code> <code>qtl.genotype.labels()</code> <code>cond.markers()</code> <code>marker.genotype.labels()</code> <code>cim.H0.regress()</code> <code>gridvals()</code> <code>em.unknown.probs()</code> <code>get.recomb()</code>
<code>validate.cross()</code>	Checks that data is valid for a particular line cross.	
<code>cofactor.matrix()</code>	Codes the matrix of marker cofactors	<code>contrasts.b1()</code> <code>contrasts.f2()</code>
<code>qtl.design()</code>	Switching function for <code>fac.design.nw()</code> to create a factorial design for the genotypes at QTL loci, based upon the breeding design and the hypothesis to be tested.	<code>fac.design.nw()</code>
<code>contrasts.b1()</code>	Returns contrasts matrices to be used in fitting the backcross model.	
<code>contrasts.f2()</code>	Returns contrasts matrices to be used in fitting the F2 model.	

Table B.1: (continued)

Function	Description	Dependencies
qtl.genotype.labels()	Creates labels (names) for the QTL genotypes.	fac.design.nw()
cond.markers()	Identifies the marker loci to condition on, depending on the model being fitted.	
marker.genotype.labels()	Creates labels (names) for the marker genotypes.	fac.design.nw()
cim.H0.regress()	Calculate, via linear regression, the maximum likelihood of the observed trait values for inbred linecross data, a null hypothesis of no QTL anywhere (model “N”).	loglik()
gridvals()	Selects starting points for the EM Algorithm.	moment.nw() checki() get.probs.start() em.known.probs()
em.unknown.probs()	Calculate (via the EM Algorithm) the maximum likelihood of the observed trait values for inbred line-cross data, assuming that the mixing proportions are not known.	mixing.probs() moment.nw() getZij mle.probs() loglik() diff.moments() haldane.probs() emcov.fisher() emcov.observed()
getZij()	Estimate QTL-category identities of individuals.	
get.recomb()	Used for informational output only.	recomb.hat.b1() recomb.hat.f2()

Table B.1: (continued)

Function	Description	Dependencies
<code>recomb.hat.b1()</code>	Calculates sample estimators of the recombination fractions markers (B1 sample). Used for informational output only.	
<code>recomb.hat.f2()</code>	Calculates sample estimators of the recombination fractions markers (F2 sample). Used for informational output only.	
<code>fac.design.nw()</code>	Used for generating all possible QTL genotypes given the number of loci and genotypes at each locus.	
<code>loglik()</code>	Calculates the natural logarithm of the mixture likelihood.	
<code>moment.nw()</code>	Calculates the k^{th} moment of a numeric vector.	
<code>checki()</code>	Reduces the number of starting possible points that are tested when selection starting values for the EM algorithm.	
<code>get.probs.start()</code>	Configures the vector of mixing parameters according to the model being fitted.	
<code>em.known.probs()</code>	Calculate (via the EM Algorithm) the maximum likelihood of the observed trait values for inbred line-cross data, for known (fixed) mixing proportions.	<code>mixing.probs()</code> <code>diff.moments()</code> <code>loglik()</code>

Table B.1: (continued)

Function	Description	Dependencies
mixing.probs()	Switching function for calculating the mixing proportions.	weights.b1() weights.f2()
diff.moments()	Calculate residual error and assess clustering of groups.	
weights.b1()	Calculate the mixing proportions for the backcross design.	index.genot()
weights.f2()	Calculate the mixing proportions for the F2 design.	index.genot()
mle.probs()	switching function to calculate the MLEs of the mixing parameters depending on the type of breeding design.	phi.hat.b1() phi.hat.f2()
phi.hat.b1()	Calculate the MLEs of the mixing parameters for the backcross design.	index.genot() constrain()
phi.hat.f2()	Calculate the MLEs of the mixing parameters for the F2 design.	index.genot constrain()
constrain()	Switching function for constrain.b1() and constrain.f2().	constrain.b1() constrain.f2()
constrain.b1()	Completes the calculation for the MLEs of mixing parameters for the backcross and ensures that they are within the valid range.	haldane.probs()
constrain.f2()	Completes the calculation for the MLEs of mixing parameters for the F2 and ensures that they are within the valid range.	haldane.probs()

Table B.1: (continued)

Function	Description	Dependencies
<code>haldane.probs()</code>	Formats the output of the MLEs of the mixing proportions.	
<code>emcov.fisher()</code>	Calculates the expected information matrix.	See Section B.4 for details.
<code>emcov.observed()</code>	Calculates the observed information matrix.	See Section B.4 for details.

Source Code

```
#-----
# rim.linecross() : Robust Interval mapping procedure for
# a sample taken from a B1,B2 or F2 population.
# PARAMETERS of rim.linecross():
# data - a data frame
# regressors - (markers)a vector of indices/names of columns in data frame,
#             the order of elements in this vector should be the same as locus order.
# all.markers - the names of all makers in the data frame given in locus order
# cross - one of "B1", "B2", "F2"
# homog.high - a character string denoting the homozygous high genotype
# heteroz - a character string denoting the heterozygous genotype
# homog.low - a character string denoting the homozygous low genotype
# hypothesis - one of "H0", "H1"
# maxit - maximum number of iterations.
# r.curr.next - vector where the ith element is the recombination frequency
#              between marker i and i+1, so the last value in r.curr.next should be 0.5,
#              markers should have the same order as given in all.markers.
# return.all - if true: returns results for all models when AIC is used
#              to select a model from among "N","R","L","LR","Q","QR","LQ","LQR"
# return.start - if true the staring values are also returned.
# tol - tolerance limit for stopping the EM algorithm.
# trait - the name of a trait in the data frame(a character string)
# validated -checks that data is valid for a particular line cross,
#            set to validated=TRUE to avoid this step when running simulations.
#-----
rim.linecross<-function(hypothesis="H1", cross, data, regressors, homog.high, heteroz, homog.low,
                        all.markers,trait, maxit=100,tol=1e-6, r.curr.next, mapfun="Haldane",validated=FALSE,
                        chosen.model="LQR", return.all=FALSE, return.start=FALSE, imat.type="expected"){

  m <- match.call()
```

```

if (!validated) #for simulations use validated=TRUE to skip this step
  validate.cross(data, regressors,homog.high, heteroz, homog.low,all.markers,trait, 2,1,cross)
gotMASS<- try(find(ginv, mode="function"),silent=TRUE)
if (inherits(gotMASS, "try-error"))
  require("MASS")
#Get ready to set up the matrix of coded cofactors
x<-names(data[,regressors])
marker.id<- pmatch(x,all.markers)
if ((marker.id[2]!=marker.id[1]+1) || (length(x)!=2))
  stop("regressors should be one pair of adjacent markers")
all.h0<-c("N","R","L","LR")
all.h1<-c("Q","QR","LQ","LQR")
allmodels<-c(all.h0,all.h1)
allhyp<-c(rep("H0",4),rep("H1",4))
names(allhyp)<-allmodels
#list the markers to condition on in each situation
nmarkers<-length(all.markers)#_MN_
if ((marker.id[1]==1)&&(marker.id[2]==nmarkers))
  stop("flanking markers are required")
else if (marker.id[1]==1) #_MNO
  amconfig<-list(N=c(K=0,0=0),R=c(K=0,0=1), L=c(K=0,0=0), LR=c(K=0,0=1),
    Q=c(K=0,0=0), QR=c(K=0,0=1), LQ=c(K=0,0=0), LQR=c(K=0,0=1))
else if (marker.id[2]==nmarkers)#KMN_
  amconfig<-list(N=c(K=0,0=0), R=c(K=0,0=0),L=c(K=1,0=0), LR=c(K=1,0=0),
    Q=c(K=0,0=0), QR=c(K=0,0=0), LQ=c(K=1,0=0), LQR=c(K=1,0=0))
else #KMNO
  amconfig<-list(N=c(K=0,0=0), R=c(K=0,0=1), L=c(K=1,0=0), LR=c(K=1,0=1),
    Q=c(K=0,0=0), QR=c(K=0,0=1), LQ=c(K=1,0=0), LQR=c(K=1,0=1))

model.id<-NULL
model1<-chosen.model
if (chosen.model=="RIM1")
  chosen.model<-switch(as.character(hypothesis),H0="LR",H1="LQR")
else if (chosen.model=="CIM")
  chosen.model<-switch(as.character(hypothesis),H0="N",H1="Q")
if (!is.null(chosen.model)){
  model.id<-pmatch(chosen.model,allmodels)
  model.id<-model.id[!is.na(model.id)]
}
nmodels<-length(model.id)
if ((nmodels==0)|| (nmodels>1)){
  model1<-"AIC"
  chosen.model<-model.desc<-"LQR"
  choice<-c("LQR","LQ","QR","Q","LR","L","R","N")
}

```

```

else{
  model.desc<-chosen.model
  choice<-chosen.model
}
nmodels<-length(choice)
if (nmodels>1)
  aic.selection<-TRUE
else
  aic.selection<-FALSE
mle.model<-as.list(1:nmodels)
names(mle.model)<-choice
hold<-mle.model
#drop any names from the vector of recombination freq
r.curr.next<-as.numeric(r.curr.next)
data<-data[!is.na(data[, trait]),]

#left.names and right.names will aid selection of starting values
left.names<-NULL
right.names<-NULL
if(marker.id[1]>1){
  left2<-cofactor.matrix(cross,homog.high,heteroz,
    homog.low,data,trait,all.markers[marker.id[1]-1])
  left.names<-dimnames(left2)[[2]] #K
}
if (marker.id[2]<nmarkers){
  right2<-cofactor.matrix(cross,homog.high,heteroz,
    homog.low,data,trait,all.markers[marker.id[2]+1])
  right.names<-dimnames(right2)[[2]] #0
}
xleft<-marker.id[1]
if (xleft>=1){
  left2<-cofactor.matrix(cross,homog.high,heteroz,
    homog.low,data,trait,all.markers[xleft])
  xleft.names<-dimnames(left2)[[2]] #M
}
xright<-marker.id[2]
if (xright<=nmarkers){
  right2<-cofactor.matrix(cross,homog.high,heteroz,
    homog.low,data,trait,all.markers[xright])
  xright.names<-dimnames(right2)[[2]] #N
}
remove(left2)
remove(right2)

#carry out model selection and/or model fitting

```

```

for(i in 1:nmodels){
  CIMHO<-FALSE
  chosen.model<-choice[i]
  hypothesis<-allhyp[chosen.model]
  mconfig<-amconfig[[chosen.model]]

  qtl.design.frame<- qtl.design(chosen.model,cross)
  if (!is.null(qtl.design.frame)){
    qtl.contrasts<-switch(as.character(cross),
      B1=lapply(qtl.design.frame,contrasts.b1,AA="QQ",Aa="Qq",hi="AA"),
      B2=lapply(qtl.design.frame,contrasts.b1,AA="qq",Aa="Qq",hi="aa"),
      F2=lapply(qtl.design.frame,contrasts.f2,AA="QQ",Aa="Qq",aa="qq",hi="AA"))
    sum.qtl<-paste(names(qtl.design.frame),collapse="+")
    # to fit all interactions use:
    #sum.qtl<-paste(names(qtl.design.frame),collapse="*")
    qtl.formula<-formula(paste("~",sum.qtl))
    qtl.modelfrm<-model.frame(qtl.formula,data=qtl.design.frame)
    Ce<-model.matrix(qtl.formula,qtl.modelfrm,contrasts=qtl.contrasts)
    dimnames(Ce)[[1]]<-qtl.genotype.labels(chosen.model,cross)
    #Ce is the qtl contrast matrix
    nqgen<-length(Ce[,1]) #number of qtl genotypes
    main.names<-dimnames(Ce)[[2]]
  }
  else {
    CIMHO<-TRUE
    Ce<-NULL
    main.names<-NULL
    nqgen<-0
  }
  #Set up the matrix of coded cofactors
  conditioning.markers<-cond.markers(mconfig,marker.id,nmarkers)
  x4<-all.markers[conditioning.markers]
  x4a<-conditioning.markers
  if (chosen.model=="R")
    conditioning.markers<-x4a[x4a!=xleft]
  else if (chosen.model=="L")
    conditioning.markers<-x4a[x4a!=xright]
  flanking.markers<-all.markers[-conditioning.markers]
  X2<-cofactor.matrix(cross,homog.high,heteroz,homog.low,data,trait,flanking.markers)
  cofactors.names<-dimnames(X2)[[2]]

  dat<-data[,c(x4,trait)]
  g<-apply(dat[,x4],1,paste,collapse="",sep="")
  N<-length(g)
  n<-tapply(g,g,length)

```



```

#markerg == (unordered) marker genotype labels from the data
markerg<-names(n)
nmgen<-length(n)
#y will be a list of trait values grouped by marker genotype
y<-split(dat[,trait],g)
indivs<-row.names(dat)
indivs<-split(indivs,g)
sorted.rows<-unlist(indivs)
MCstar<-X2[sorted.rows,]
dimnames(MCstar)[[1]]<-1:N
remove(list=c("dat","X2"))
#store properties to describe the data structure of the current model
rmn<-r.curr.next[marker.id[1]]
if (marker.id[1]>1)
  marker.map<-r.curr.next[c(marker.id[1]-1,marker.id)] #rlm,rmn,rnr
else marker.map<-c(0.5,r.curr.next[marker.id]) #rlm,rmn,rnr
rlm<-marker.map[1]
rnr<-marker.map[3]
genot<-marker.genotype.labels(cross,homog.high, heteroz,
  homog.low,FALSE,mconfig)
genot2<-marker.genotype.labels(cross,homog.high, heteroz,
  homog.low,TRUE,mconfig)
fullqtl<-qtl.genotype.labels("LQR",cross)
genot2<-list(g=genot2,mconfig=mconfig,rmn=rmn,rno=rnr,rkm=rlm,
  hi=homog.high,het=heteroz, low=homog.low,qtl3=fullqtl)
hold[[i]]<-list(hypothesis=hypothesis,Ce=Ce,MCstar=MCstar,n=n,y=y, nqgen=nqgen,genot=genot,
  genot2=genot2,markerg=markerg,cofactors.names=cofactors.names,mconfig=mconfig,
  flanking.markers=flanking.markers,x4=x4)
#now do parameter estimation by finding the mle
if(CIMHO==TRUE){
  mle<- cim.H0.regress(MCstar,cofactors.names,n,y,genot,genot2,startvals=FALSE)
  startvals<-list(desc="No QTL Anywhere: Marker Regression only")
  mle.model[[i]] <-list(mle=mle,startvals=startvals)
}
else{
  #identify the cofactor categories within marker categories to help
  #in selecting starting values for the EM algorithm.
  ind<-vector("list", nmgen)
  names(ind)<-names(n)
  ind[[1]]<-1:n[1]
  for(j in 2:nmgen)
    ind[[j]]<- (1+sum(n[1:(j-1)])):sum(n[1:j])
  yc<-as.list(n)
  nc<-indc<-as.list(n)
  for(j in 1:nmgen){

```

```

      Mj<-rbind(MCstar[ind[[j]], ])
      ge<-apply(Mj,1,paste,collapse="",sep="")
      yc[[j]]<-split(y[[j]],ge)
      nc[[j]]<-sapply(yc[[j]],length)
      names(nc[[j]])<-NULL
      indivsc<-1:n[j]
      if (length(ge)>1)
        indc[[j]]<-split(indivsc,ge)
      else
        indc[[j]]<-list(indivsc)
    }
    #get starting values that reduce residual error while
    #separating groups
    startvals<-gridvals(cross,hypothesis,Ce,MCstar,cofactors.names,
                        mapfun,n,nqgen,y,genot,genot2,chosen.model,ind,indc,nc,yc)
    probs.start<-startvals$model.params$probs
    B.start<-startvals$model.params$effects
    sigma2.start<-startvals$model.params$variance
    recomb.start<-startvals$recomb
    #compute the maximum likelihood via the EM algorithm
    mle<-em.unknown.probs(chosen.model,cross,hypothesis,Ce,MCstar,cofactors.names,
                        sigma2.start, B.start,probs.start,tol,maxit,mapfun,n,nqgen,y,genot,genot2,
                        startvals=FALSE,recomb.start,imat.type,startvals$mvars,ind,indc,nc,yc)
    #store vlaues to output
    if (return.start==T)
      mle.model[[i]] <-list(mle=mle,startvals=startvals)
    else
      mle.model[[i]] <-list(mle=mle)
  }
  if (aic.selection)
    mle.model[[i]]$mle$AIC<- (-2*mle.model[[i]]$mle$loglike
      +2*(length(mle.model[[i]]$mle$model.params$effects[,1])
      +length(mle.model[[i]]$mle$model.params$probs[,1])) )
  if (i==1){
    bestmodel<-i
    if (aic.selection)
      minAIC<- mle.model[[i]]$mle$AIC
  }
  else if (aic.selection){
    if (mle.model[[i]]$mle$AIC<=minAIC){
      bestmodel<-i
      minAIC<-mle.model[[i]]$mle$AIC
    }
  }
} #end for(i in 1:nmodels) ....

```

```

hold<-hold[[bestmodel]]
chosen.model<-as.character(choice[bestmodel])
#FINISHED
#print some information to assess the quality of the sample
#what are the sample estimates of rkm, rmn,rno?
genotLQR<-marker.genotype.labels(cross,homog.high, heteroz,homog.low,FALSE,amconfig$LQR)
genot2LQR<-marker.genotype.labels(cross,homog.high, heteroz,homog.low,TRUE,amconfig$LQR)
genot2LQR<-list(g=genot2LQR,mconfig=amconfig$LQR,rmn=rmn,rno=rnr,rkm=rlm,
               hi=homog.high,het=heteroz, low=homog.low,qt13=fullqt1)
condLQR.markers<-cond.markers(amconfig$LQR,marker.id,nmarkers)
x4LQR<-all.markers[condLQR.markers]
datLQR<-data[,c(x4LQR,trait)]
gLQR<-apply(datLQR[,x4LQR],1,paste,collapse="",sep="")
remove(datLQR)
nLQR<-tapply(gLQR,gLQR,length)
markergLQR<-names(nLQR)
map.hat<-get.recomb(cross,nLQR,genotLQR,genot2LQR,markergLQR)

#format the output
obj.name<-strsplit(deparse(m$data), "[$,]")[[1]][1]
obj.name<-as.name(obj.name)
data<-strsplit(deparse(m$data), "[$,]")[[1]][2]
val<-list(code=deparse(m[1]),information.matrix=imat.type,
          chosen.model.desc=model1,chosen.model=choice[bestmodel],mapfun=mapfun,
          cross=cross,hypothesis=as.character(allhyp[choice[bestmodel]]),
          data=list(obj.name=obj.name,dat=data),interval=x,markers=hold$x4,
          extra.markers=hold$flanking.markers,trait=trait,
          genotype.counts=hold$n, map.hat=map.hat,mle=mle.model[[bestmodel]]$mle)
if (return.start==TRUE){
  val$startvals.best<-mle.model[[bestmodel]]$startvals
}
if ((aic.selection==TRUE) && (return.all==TRUE))
  val$all<-mle.model

val
}

#-----
# validate.cross() : checks that data is valid for a particular line cross
#-----
validate.cross <- function(data, regressors,homog.high, heteroz, homog.low,
                           all.markers,trait, nfactors=2,ntraits=1,cross){

  if(!is.data.frame(data))
    stop(paste(m$data, "should be of mode data.frame"))
  if(!is.vector(regressors))

```

```

    stop("regressors should be a vector")
  if(!is.character(trait))
    stop("trait should be a character string")
  nregs<-length(regressors)
  if (nregs!=nfactors)
    stop(paste("this method is designed for",nfactors,"loci only"))
  if (length(trait)!=ntraits)
    stop(paste("Expected one response variable, found",length(trait)))
  if(length(unique(c(homog.high,heteroz,homog.low)))!=3)
    stop("homog.high, heteroz and homog.low should be unique.")

  x<-names(data[,regressors])
  if(length(unique(x))!=nfactors)
    stop("Regressors should be unique.")
  dat<-data.frame(data[!is.na(data[, trait]),x])
  names(dat)<-x
  fac<-lapply(dat, is.factor)
  fac<-unlist(fac)
  if(length(fac[fac==T])!=length(x) )
    stop("Regressors must be factors.")

  fac<-lapply(dat, function(h){any(is.na(h))})
  fac<-unlist(fac)
  if(length(fac[fac==T])>0 )
    stop(paste("Missing values are not allowed at the markers.",
              "Missing values found in",paste(names(fac[fac==T]),collapse=",")))
  levs<-lapply(dat,levels)
  bad<-function(h,AA,Aa,aa,type){
    nbad<-switch(as.character(type),
      B1=length(h[(h!=AA) &(h!=Aa)]),
      B2=length(h[(h!=Aa) &(h!=aa)]),
      F2=length(h[(h!=AA) &(h!=Aa) &(h!=aa)])
    )
    nbad
  }
  numbad<-lapply(levs,bad,AA=homog.high, Aa=heteroz,aa=homog.low,type=cross)
  numbad<-unlist(numbad)
  if (length(numbad[numbad!=0])!=0)
    stop(paste("Invalid", cross, "data"))
}

#-----
# contrasts.b1(), contrasts.f2() :
# These functions return contrast matrices for extracting
# certain linear combinations of marker/qlt effects in B1, B2 and F2 samples.

```

```

#
# PARAMETERS of contrasts.b1(), contrasts.f2():
# h - A data frame (or list) containing marker genotypes
# AA - a character string denoting the homozygous high genotype
# Aa - a character string denoting the heterozygous genotype
# aa - a character string denoting the homozygous low genotype
# hi - character string indicating the 'high' genotype
#-----
contrasts.b1<-function(h,AA,Aa,hi){
  genotypes<-levels(h)
  cmat<-matrix(nrow=2,ncol=1,
    dimnames=list(genotypes,paste(".",hi,sep="")))
  cmat[as.character(AA),]<- 1
  cmat[as.character(Aa),]<- 0
  cmat
}#end of contrasts.b1()

contrasts.f2<-function(h,AA,Aa,aa,hi){
  genotypes<-levels(h)
  cmat<-matrix(nrow=3,ncol=2,
    dimnames=list(genotypes,paste(c(".a",".d"),hi,sep="")))
  cmat[as.character(AA),]<-c(1,1)
  cmat[as.character(Aa),]<-c(0,-1)
  cmat[as.character(aa),]<-c(-1,1)
  cmat
}

#-----
# fac.design.nw():
# Generates simple factorial design (does not support fractional factorial designs and replications).
# Used for generating all possible QTL genotypes
# given the number of loci and genotypes at each locus.
# This function is adapted from the S-PLUS function fac.design().
# Example of use:
# fnames<-list(L=c("LL","Ll","ll"),Q=c("QQ","Qq","qq"),R=c("RR","Rr","rr"))
# y <- fac.design.nw(rep(3,3), factor.names = fnames)
#-----
fac.design.nw<-function(levels, factor.names){
  if(any(is.na(levels)) || any(as.integer(levels) - levels != 0))
    stop("levels must be integer and positive")
  nrows <- prod(levels)
  ncols <- length(levels)
  if(ncols && nrows > 1000000.)
    cat("Attempting to create a design with", nrows, "rows\n")
  yy <- as.list(1:ncols)

```

```

if(ncols==1)
  yy[[1]]<-factor.names
else{
  rep1 <- prod(levels[1:(ncols-1)]) #sort in a top-down manner(unlike S-PLUS)
  for(i in 1:ncols) {
    lev <- 1:levels[i]
    j <- rep(rep(lev, rep(rep1, levels[i])), length = nrows)
    yy[[i]]<-factor.names[[i]][j]
    yy[[i]] <- as.factor(yy[[i]])
    rep1 <- rep1/levels[i]
  }
}
names(yy)<-names(factor.names)
yy<-data.frame(yy)
yy
}

#-----
# qtl.design(): Switching function for fac.design.nw() to createa factorial design for the genotypes
# at QTL loci, based upon the breeding design and the hypothesis to be tested.
#-----

qtl.design<- function(chosen.model,cross){
  gb1<-c("QQ","Qq")
  gb2<-c("qq","Qq")
  gf2<-c("QQ","Qq","qq")
  qdesign<-switch(as.character(cross),
    B1=switch(as.character(chosen.model),
      N=NULL,
      R=fac.design.nw(rep(2,1), list(R=gb1)),
      L=fac.design.nw(rep(2,1), list(L=gb1)),
      LR=fac.design.nw(rep(2,2), list(L=gb1,R=gb1)),
      Q=fac.design.nw(rep(2,1), list(Q=gb1)),
      QR=fac.design.nw(rep(2,2), list(Q=gb1,R=gb1)),
      LQ=fac.design.nw(rep(2,2), list(L=gb1,Q=gb1)),
      LQR=fac.design.nw(rep(2,3), list(L=gb1,Q=gb1,R=gb1))
    ),
    B2=switch(as.character(chosen.model),
      N=NULL,
      R=fac.design.nw(rep(2,1), list(R=gb2)),
      L=fac.design.nw(rep(2,1), list(L=gb2)),
      LR=fac.design.nw(rep(2,2), list(L=gb2,R=gb2)),
      Q=fac.design.nw(rep(2,1), list(Q=gb2)),
      QR=fac.design.nw(rep(2,2), list(Q=gb2,R=gb2)),
      LQ=fac.design.nw(rep(2,2), list(L=gb2,Q=gb2)),

```

```

        LQR=fac.design.nw(rep(2,3), list(L=gb2,Q=gb2,R=gb2))
    ),
    F2=switch(as.character(chosen.model),
        N=NULL,
        R=fac.design.nw(rep(3,1), list(R=gf2)),
        L=fac.design.nw(rep(3,1), list(L=gf2)),
        LR=fac.design.nw(rep(3,2), list(L=gf2,R=gf2)),
        Q=fac.design.nw(rep(3,1), list(Q=gf2)),
        QR=fac.design.nw(rep(3,2), list(Q=gf2,R=gf2)),
        LQ=fac.design.nw(rep(3,2), list(L=gf2,Q=gf2)),
        LQR=fac.design.nw(rep(3,3), list(L=gf2,Q=gf2,R=gf2))
    )
)

qdesign
}

#-----
# marker.genotype.labels()
#-----
marker.genotype.labels<-function(cross,homog.high, heteroz, homog.low,index=FALSE,mconfig){
  gb1<-c(homog.high,heteroz)
  gb2<-c(homog.low,heteroz)
  gf2<-c(homog.high,heteroz,homog.low)
  if ((mconfig["K"]==0)&& (mconfig["0"]==0)){
    labels<-switch(as.character(cross),
      B1=fac.design.nw(rep(2,2), list(M=gb1,N=gb1)),
      B2=fac.design.nw(rep(2,2), list(M=gb2,N=gb2)),
      F2=fac.design.nw(rep(3,2), list(M=gf2,N=gf2)))
  }
  else if ((mconfig["K"]==0)&& (mconfig["0"]==1)){
    labels<-switch(as.character(cross),
      B1=fac.design.nw(rep(2,3), list(M=gb1,N=gb1,O=gb1)),
      B2=fac.design.nw(rep(2,3), list(M=gb2,N=gb2,O=gb2)),
      F2=fac.design.nw(rep(3,3), list(M=gf2,N=gf2,O=gf2)))
  }
  else if ((mconfig["K"]==1)&& (mconfig["0"]==0)){
    labels<-switch(as.character(cross),
      B1=fac.design.nw(rep(2,3), list(K=gb1,M=gb1,N=gb1)),
      B2=fac.design.nw(rep(2,3), list(K=gb2,M=gb2,N=gb2)),
      F2=fac.design.nw(rep(3,3), list(K=gf2,M=gf2,N=gf2)))
  }
  else {
    labels<-switch(as.character(cross),

```

```

        B1=fac.design.nw(rep(2,4), list(K=gb1,M=gb1,N=gb1,O=gb1)),
        B2=fac.design.nw(rep(2,4), list(K=gb2,M=gb2,N=gb2,O=gb2)),
        F2=fac.design.nw(rep(3,4), list(K=gf2,M=gf2,N=gf2,O=gf2)))
    }
    if(index==TRUE){
        val<-as.data.frame(labels)
        val<-cbind(val,index=1:length(labels[,1]))
    }
    else{
        val<-apply(labels,1,paste,collapse="",sep="")
        names(val)<-val
    }
    val
}

```

```

#-----
# qtl.genotype.labels()
#-----

qtl.genotype.labels<- function(chosen.model,cross){
    gb1Q<-c("QQ","Qq")
    gb2Q<-c("qq","Qq")
    gf2Q<-c("QQ","Qq","qq")
    gb1L<-c("LL","Ll")
    gb2L<-c("ll","Ll")
    gf2L<-c("LL","Ll","ll")
    gb1R<-c("RR","Rr")
    gb2R<-c("rr","Rr")
    gf2R<-c("RR","Rr","rr")

    labels<-switch(as.character(cross),
        B1=switch(as.character(chosen.model),
            N=NULL,
            R=fac.design.nw(rep(2,1), list(R=gb1R)),
            L=fac.design.nw(rep(2,1), list(L=gb1L)),
            LR=fac.design.nw(rep(2,2), list(L=gb1L,R=gb1R)),
            Q=fac.design.nw(rep(2,1), list(Q=gb1Q)),
            QR=fac.design.nw(rep(2,2), list(Q=gb1Q,R=gb1R)),
            LQ=fac.design.nw(rep(2,2), list(L=gb1L,Q=gb1Q)),
            LQR=fac.design.nw(rep(2,3), list(L=gb1L,Q=gb1Q,R=gb1R))
        ),
        B2=switch(as.character(chosen.model),
            N=NULL,
            R=fac.design.nw(rep(2,1), list(R=gb2R)),
            L=fac.design.nw(rep(2,1), list(L=gb2L)),
            LR=fac.design.nw(rep(2,2), list(L=gb2L,R=gb2R)),

```



```

        Q=fac.design.nw(rep(2,1), list(Q=gb2Q)),
        QR=fac.design.nw(rep(2,2), list(Q=gb2Q,R=gb2R)),
        LQ=fac.design.nw(rep(2,2), list(L=gb2L,Q=gb2Q)),
        LQR=fac.design.nw(rep(2,3), list(L=gb2L,Q=gb2Q,R=gb2R))
    ),
    F2=switch(as.character(chosen.model),
        N=NULL,
        R=fac.design.nw(rep(3,1), list(R=gf2R)),
        L=fac.design.nw(rep(3,1), list(L=gf2L)),
        LR=fac.design.nw(rep(3,2), list(L=gf2L,R=gf2R)),
        Q=fac.design.nw(rep(3,1), list(Q=gf2Q)),
        QR=fac.design.nw(rep(3,2), list(Q=gf2Q,R=gf2R)),
        LQ=fac.design.nw(rep(3,2), list(L=gf2L,Q=gf2Q)),
        LQR=fac.design.nw(rep(3,3), list(L=gf2L,Q=gf2Q,R=gf2R))
    )
)
)
if (is.null(labels))
    val<-NULL
else{
    val<-apply(labels,1,paste,collapse="",sep="")
    names(val)<-val
}
val
}

#-----
# cofactor.matrix()
#-----

cofactor.matrix<-function(cross,homog.high,heteroz,homog.low,data,trait,flanking.markers){
    marker.design.frame<-as.data.frame(data[,flanking.markers])
    names(marker.design.frame)<-flanking.markers
    marker.contrasts<-switch(as.character(cross),
        B1=lapply(marker.design.frame,contrasts.b1,AA=homog.high, Aa=heteroz,hi=homog.high),
        B2=lapply(marker.design.frame,contrasts.b1,AA=homog.low, Aa=heteroz,hi=homog.low),
        F2=lapply(marker.design.frame,contrasts.f2,AA=homog.high,
            Aa=heteroz,aa=homog.low,hi=homog.high)
    )
    sum.flank<-paste(flanking.markers,collapse="+")
    cofactors.formula<-formula(paste("~",sum.flank))
    #Set up the matrix of coded cofactors
    modelfrm<-model.frame(cofactors.formula,data=data)
    X2<-model.matrix(cofactors.formula,modelfrm,contrasts=marker.contrasts)
    xnames<-dimnames(X2)[[2]]
    X2<-cbind(X2,-1)
}

```

```

        dimnames(X2)[[2]]<-xnames[-1]
        X2
    }

#-----
# cond.markers()
#-----
cond.markers<-function(mconfig,marker.id,nmarkers){
    if ((mconfig["K"]==0)&& (mconfig["0"]==0))
        marker.id2<-marker.id
    else if ((mconfig["K"]==1)&& (mconfig["0"]==0))
        marker.id2<-c(marker.id[1]-1,marker.id)
    else if ((mconfig["K"]==0)&& (mconfig["0"]==1))
        marker.id2<-c(marker.id,marker.id[2]+1)
    else
        marker.id2<-c(marker.id[1]-1,marker.id,marker.id[2]+1)
    marker.id2<-marker.id2[(marker.id2>=1)&(marker.id2<=nmarkers)]
    marker.id2
}

#-----
# recomb.hat.b1() : used for assessing the quality of the sample. Used for informational output only.
# Not used for model fitting.
# To see if recombination frequencies between markers, as estimated from the
# sample, resemble recombination frequencies between markers in the assumed marker map.
#-----
recomb.hat.b1<-function(n,genot,genot2,markerg){
    cross<-"B1"

    #sort marker genotype counts by marker category
    n<-n[genot]
    temp<-n
    n<-rep(0,length(genot))
    names(n)<-genot
    n[markerg]<-temp[markerg]

    N<-sum(n)

    mconfig<-genot2$mconfig
    gmn<-index.genot(cross="B1",M="M",N="N",genot2=genot2)
    rMN<-sum(n[c(gmn$MMNn,gmn$MmNN)])/N
    #rMN<-min(genot2$rmn,rMN)

    if (mconfig["K"]==1){
        gkm<-index.genot(cross="B1",M="K",N="M",genot2=genot2)
    }
}

```

```

        rKM<-sum(n[c(gkm$KKMm,gkm$KkMM)])/N
        # rKM<-min(genot2$rkm,rKM)

    }
    else
        rKM<-NA #genot2$rkm
    if (mconfig["O"]==1){
        geno<-index.genot(cross="B1",M="N",N="O",genot2=genot2)
        rNO<-sum(n[c(gno$NNOo,gno$NnOO)])/N
    }
    else
        rNO<- NA #genot2$rno
    map.hat<-c(rKM=rKM,rMN=rMN,rNO=rNO)
    map.hat<-map.hat[!is.na(map.hat)]
    map.hat
}

#-----
# recomb.hat.f2() : used for assessing the quality of the sample.
# Used for informational output only. Not used for model fitting.
# To see if recombination frequencies between markers, as estimated from the
# sample, resemble recombination frequencies between markers in the assumed marker map.
#-----
recomb.hat.f2<-function(n,genot,genot2,markerg){
    cross<-"F2"
    #sort marker genotype counts by marker category
    n<-n[genot]
    temp<-n
    n<-rep(0,length(genot))
    names(n)<-genot
    n[markerg]<-temp[markerg]
    N<-sum(n)
    mconfig<-genot2$mconfig
    gmn<-index.genot(cross=cross,M="M",N="N",genot2=genot2)

    rMN<- (1- (2*sum(n[gmn$mmnn])/N + 2*sum(n[gmn$MMNN])/N
        + sum(n[gmn$mmNn])/N + sum(n[gmn$MMNn])/N))

    if (mconfig["K"]==1){
        gkm<-index.genot(cross=cross,M="K",N="M",genot2=genot2)
        rKM<- (1- (2*sum(n[gkm$kkmm])/N + 2*sum(n[gkm$KKMM])/N
            + sum(n[gkm$kkMm])/N + sum(n[gkm$KKMm])/N))
    }
    else
        rKM<-NA #genot2$rkm

```

```

if (mconfig["0"]==1){
  gno<-index.genot(cross=cross,M="N",N="0",genot2=genot2)
  rNO<- (1- (2*sum(n[gno$nnoo])/N + 2*sum(n[gno$NN00])/N
            + sum(n[gno$nn0o])/N + sum(n[gno$NN0o])/N))
}
else
  rNO<- NA #genot2$rno
map.hat<-c(rKM=rKM,rMN=rMN,rNO=rNO)
map.hat<-map.hat[!is.na(map.hat)]
map.hat
}

#-----
# get.recomb() : used for assessing the quality of the sample.
# Used for informational output only. Not used for model fitting.
#-----
get.recomb<-function(cross,...){
  recomb.hat<-switch(as.character(cross),
    B1=recomb.hat.b1(...),
    B2=recomb.hat.b1(...),
    F2= recomb.hat.f2(...))
  recomb.hat
}

#-----
# index.genot.b1() : identify one- or two-locus marker genotypes in the model.
#-----
index.genot.b1<-function(M=NULL,N=NULL,genot2){
  mconfig<-genot2$mconfig
  if ((!is.null(M))&&(!is.null(N))){
    MMNN<-genot2$g$index[(genot2$g[,M]==genot2$hi)&(genot2$g[,N]==genot2$hi)]
    MMNn<-genot2$g$index[(genot2$g[,M]==genot2$hi)&(genot2$g[,N]==genot2$het)]
    MmNN<-genot2$g$index[(genot2$g[,M]==genot2$het)&(genot2$g[,N]==genot2$hi)]
    MmNn<-genot2$g$index[(genot2$g[,M]==genot2$het)&(genot2$g[,N]==genot2$het)]
    val<-list(MMNN,MMNn,MmNN,MmNn)
    names(val)<-switch(M,
      K=c("KKMM","KKMm","KkMM","KkMm"),
      M=c("MMNN","MMNn","MmNN","MmNn"),
      N=c("NN00","NNOo","Nn00","NnOo"))
  }
  else if (is.null(N)){
    MM<-genot2$g$index[(genot2$g[,M]==genot2$hi)]
    Mm<-genot2$g$index[(genot2$g[,M]==genot2$het)]
    val<-list(MM,Mm)
    names(val)<-c("MM","Mm")
  }
}

```

```

}
else{
  NN<-genot2$g$index[(genot2$g[,N]==genot2$hi)]
  Nn<-genot2$g$index[(genot2$g[,N]==genot2$het)]
  val<-list(NN,Nn)
  names(val)<-c("NN","Nn")
}
val
}

#-----
# index.genot.f2() : identify one- or two-locus marker genotypes in the model.
#-----

index.genot.f2<-function(M=NULL,N=NULL,genot2){
  mconfig<-genot2$mconfig
  if ((!is.null(M))&&(!is.null(N))){
    MMNN<-genot2$g$index[(genot2$g[,M]==genot2$hi)&(genot2$g[,N]==genot2$hi)]
    MMNn<-genot2$g$index[(genot2$g[,M]==genot2$hi)&(genot2$g[,N]==genot2$het)]
    MMnn<-genot2$g$index[(genot2$g[,M]==genot2$hi)&(genot2$g[,N]==genot2$low)]
    MmNN<-genot2$g$index[(genot2$g[,M]==genot2$het)&(genot2$g[,N]==genot2$hi)]
    MmNn<-genot2$g$index[(genot2$g[,M]==genot2$het)&(genot2$g[,N]==genot2$het)]
    Mmnn<-genot2$g$index[(genot2$g[,M]==genot2$het)&(genot2$g[,N]==genot2$low)]
    mmNN<-genot2$g$index[(genot2$g[,M]==genot2$low)&(genot2$g[,N]==genot2$hi)]
    mmNn<-genot2$g$index[(genot2$g[,M]==genot2$low)&(genot2$g[,N]==genot2$het)]
    mmnn<-genot2$g$index[(genot2$g[,M]==genot2$low)&(genot2$g[,N]==genot2$low)]
    val<-list(MMNN,MMNn,MMnn,MmNN,MmNn,Mmnn,mmNN,mmNn,mmnn)
    names(val)<-switch(M,
      K=c("KKMM","KKMm","KKmm","KkMM","KkMm","Kkmm","kkMM","kkMm","kkmm"),
      M=c("MMNN","MMNn","MMnn","MmNN","MmNn","Mmnn","mmNN","mmNn","mmnn"),
      N=c("NN00","NNOo","NNoo","Nn00","NnOo","Nnoo","nn00","nnOo","nnoo"))
  }
  else if (is.null(M)){
    NN<-genot2$g$index[(genot2$g[,N]==genot2$hi)]
    Nn<-genot2$g$index[(genot2$g[,N]==genot2$het)]
    nn<-genot2$g$index[(genot2$g[,N]==genot2$low)]
    val<-list(NN,Nn,nn)
    names(val)<-c("NN","Nn","nn")
  }
  else{
    MM<-genot2$g$index[(genot2$g[,M]==genot2$hi)]
    Mm<-genot2$g$index[(genot2$g[,M]==genot2$het)]
    mm<-genot2$g$index[(genot2$g[,M]==genot2$low)]
    val<-list(MM,Mm,mm)
    names(val)<-c("MM","Mm","mm")
  }
}

```

```

val
}
#-----
# index.genot() : identify one- or two-locus marker genotypes in the model.
#-----
index.genot<-function(cross,M=NULL,N=NULL,genot2){
  gind<-switch(as.character(cross),
    B1=index.genot.b1(M,N,genot2),
    B2=index.genot.b2(M,N,genot2),
    F2=index.genot.f2(M,N,genot2))
  gind
}

#-----
# get.probs.start() : configure the starting mixing proportions depending on the model being fitted.
#-----
get.probs.start<-function(chosen.model,p,mconfig){
  if ((mconfig["K"]==1)&& (mconfig["0"]==1)){
    probs.start<-switch(as.character(chosen.model),
      N=NULL,
      R=c(pR1=p$pR1,pR2=p$pR2),
      L=c(pL1=p$pL1,pL2=p$pL2),
      LR=c(pL1=p$pL1,pL2=p$pL2,pR1=p$pR1,pR2=p$pR2),
      Q=c(pQ1=p$pQ1,pQ2=p$pQ2),
      QR=c(pQ1=p$pQ1,pQ2=p$pQ2,pR1=p$pR1,pR2=p$pR2),
      LQ= c(pL1=p$pL1,pL2=p$pL2,pQ1=p$pQ1,pQ2=p$pQ2),
      LQR= c(pL1=p$pL1,pL2=p$pL2,pQ1=p$pQ1,pQ2=p$pQ2,pR1=p$pR1,pR2=p$pR2))
  }
  else if ((mconfig["K"]==0)&& (mconfig["0"]==1)){
    probs.start<-switch(as.character(chosen.model),
      N=NULL,
      R=c(pR1=p$pR1,pR2=p$pR2),
      L=c(pL=p$pL),
      LR=c(pL=p$pL,pR1=p$pR1,pR2=p$pR2),
      Q=c(pQ1=p$pQ1,pQ2=p$pQ2),
      QR=c(pQ1=p$pQ1,pQ2=p$pQ2,pR1=p$pR1,pR2=p$pR2),
      LQ= c(pL=p$pL,pQ1=p$pQ1,pQ2=p$pQ2),
      LQR= c(pL=p$pL,pQ1=p$pQ1,pQ2=p$pQ2,pR1=p$pR1,pR2=p$pR2))
  }
  else if ((mconfig["K"]==1)&& (mconfig["0"]==0)){
    probs.start<-switch(as.character(chosen.model),
      N=NULL,
      R=c(pR=p$pR),
      L=c(pL1=p$pL1,pL2=p$pL2),
      LR=c(pL1=p$pL1,pL2=p$pL2,pR=p$pR),

```

```

        Q=c(pQ1=p$pQ1,pQ2=p$pQ2),
        QR=c(pQ1=p$pQ1,pQ2=p$pQ2,pR=p$pR),
        LQ= c(pL1=p$pL1,pL2=p$pL2,pQ1=p$pQ1,pQ2=p$pQ2),
        LQR= c(pL1=p$pL1,pL2=p$pL2,pQ1=p$pQ1,pQ2=p$pQ2,pR=p$pR))
    }
else { #((mconfig["K"]==0)&& (mconfig["O"]==0))
    probs.start<-switch(as.character(chosen.model),
        N=NULL,
        R=c(pR=p$pR),
        L=c(pL=p$pL),
        LR=c(pL=p$pL,pR=p$pR),
        Q=c(pQ1=p$pQ1,pQ2=p$pQ2),
        QR=c(pQ1=p$pQ1,pQ2=p$pQ2,pR=p$pR),
        LQ= c(pL=p$pL,pQ1=p$pQ1,pQ2=p$pQ2),
        LQR= c(pL=p$pL,pQ1=p$pQ1,pQ2=p$pQ2,pR=p$pR))
    }

    probs.start
}

#-----
# checki() : speed up selection of starting values and reduce overspecification.
#-----
checki<-function(yesL, yesQ, yesR, p){
#control the possibility of overspecifation by ensuring that
#each grid test point has at most one QTL away from a marker
#when testing starting values.
k9<-0.999
k0<-1e-3
    if (yesL && yesQ && yesR){
        val<- ( ((p$pL2>=k9) && (p$pQ2>=k9))
            || ((p$pL2>=k9) && (p$pR2<=k0))
            || ((p$pQ2<=k0) && (p$pR2<=k0)) )
    }
    else if (yesQ && yesL )
        val<- ((p$pL2>=k9) || (p$pQ2<=k0))
    else if (yesQ && yesR)
        val<- ((p$pQ2>=k9) || (p$pR2<=k0))
    else val<-TRUE

    val
}

#-----
# gridvals() : to get good starting point for the EM Algorithm,

```

```
#      lay down a grid to determine "trial" mixing proportions.
#      The chosen starting values will be the point reduces
#      residual error while separating groups, as measured
#      by the variable diff returned by the function diff.moments()
#-----
```

```
gridvals<- function(cross,hypothesis,Ce,MCstar,cofactors.names,
                    mapfun,n,nqgen,y,genot,genot2,chosen.model,
                    ind,indc,nc,yc){
```

```
  #lay out the grid
  mconfig<-genot2$mconfig
  rmn<-genot2$rmn
  rkm<-genot2$rkm
  rno<-genot2$rno
```

```
  Qfit<-grep("Q",chosen.model)
  Rfit<-grep("R",chosen.model)
  Lfit<-grep("L",chosen.model)
```

```
  yesL<-(length(Lfit)>0)
  yesQ<-(length(Qfit)>0)
  yesR<-(length(Rfit)>0)
```

```
  pp<-seq(1e-5,1-1e-5,(1-2e-5)/20)
  pp[pp==0.5]<-0.5+1e-5
  pL.vals<-pQ.vals<-pR.vals<-0
  pL.vals<-0
  pQ.vals<-0
  pR.vals<-0
  if (yesL)
    pL.vals<-pp
  if (yesQ)
    pQ.vals<-pp
  if (yesR)
    pR.vals<-pp
```

```
  len.pR<-length(pR.vals)
  len.pL<-length(pL.vals)
  len.pQ<-length(pQ.vals)
```

```
  nmgen<-length(n)
  markerg<-names(y) #names of the marker genotypes
  N<-sum(n)
  #We need an index to identify the marker groups
```



```

Y<-unlist(y)
yvar<-moment.nw(Y,2)

mixing.params<-as.list(NULL)
recomb<-as.list(NULL)
maxlike<- -Inf
sdiff<- Inf
for (i in 1:len.pQ){
  if (yesQ){
    pQ2<-pQ.vals[i]
    pQ1<-0.5 + 0.5/(1-rmn)*sqrt(1-2*rmn+rmn^2*(1-2*pQ2)^2)
    mixing.params$pQ1<-pQ1 #P(QQ|MMNN)
    mixing.params$pQ2<-pQ2 #P(QQ|MMNn)
    recomb$rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
    recomb$rQN<-(1-rmn)*(1-pQ1) + rmn*pQ2
    if (mapfun=="Haldane")
      recomb$pQ1<-mixing.params$pQ1
  }
  for (il in 1:len.pL){
    if (yesL){
      pL2<-pL.vals[il]
      pL1<-0.5 + 0.5/(1-rkm)*sqrt(1-2*rkm+rkm^2*(1-2*pL2)^2)
      mixing.params$pL1<-pL1
      mixing.params$pL2<-pL2
      recomb$rKL<-(1-rkm)*(1-pL1) + rkm*(1-pL2)
      recomb$rLM<-(1-rkm)*(1-pL1) + rkm*pL2
      mixing.params$pL<-(1-recomb$rLM)
      if (mapfun=="Haldane")
        recomb$pL1<-mixing.params$pL1
      recomb$rKM<-rkm
    }
    for (ir in 1:len.pR){
      if (yesR){
        pR2<-pR.vals[ir]
        pR1<-0.5 + 0.5/(1-rno)*sqrt(1-2*rno+rno^2*(1-2*pR2)^2)
        mixing.params$pR1<-pR1
        mixing.params$pR2<-pR2
        recomb$rNR<-(1-rno)*(1-pR1) + rno*(1-pR2)
        recomb$rRO<-(1-rno)*(1-pR1) + rno*pR2
        mixing.params$pR<-(1-recomb$rNR)
        if (mapfun=="Haldane")
          recomb$pR1<-mixing.params$pR1
        recomb$rNO<-rno
      }
      recomb$rMN<-rmn
    }
  }
}

```

```

    recomb2<-unlist(recomb)
    names(recomb2)<-names(recomb)
    ok.probs<-checki(yesL, yesQ, yesR, mixing.params)

    if (ok.probs){

      probs.start<-get.probs.start(chosen.model,
        mixing.params, mconfig)
      mle1<-em.known.probs(chosen.model, cross, hypothesis,
        Ce, MCstar, cofactors.names, probs.start,
        mapfun, rmn, n, nqgen, y, genot, genot2,
        recomb2, nmgen, N, markerg, yvar, ind, indc, nc, yc)

      #if (mle1$loglike>=maxlike){
      if (mle1$mvars$diff<=sdiff){
        # maxlike<-mle1$loglike
        sdiff<- mle1$mvars$diff
        model.start<-mle1
      }
    }
  } #end for ir
} #end for il
} #end for i

probs.start<-model.start$model.params$probs
recomb.start<-model.start$recomb
startvals<-em.known.probs(chosen.model, cross, hypothesis,
  Ce, MCstar, cofactors.names, probs.start,
  mapfun, rmn, n, nqgen, y, genot, genot2,
  recomb.start, nmgen, N, markerg, yvar,
  ind, indc, nc, yc, calclike=TRUE)

startvals
}

#-----
# weights.b1() : calculate the mixing proportions for a B1 cross
#-----
weights.b1<- function(chosen.model, probs, markerg, genot, genot2, nqgen){
  Qfit<-grep("Q", chosen.model)
  Rfit<-grep("R", chosen.model)
  Lfit<-grep("L", chosen.model)
  xgen<-NULL

```

```

qt13<-genot2$qt13
highL<-grep("LL",qt13)
highR<-grep("RR",qt13)
highQ<-grep("QQ",qt13)

ppl<-matrix(1,nrow=8,ncol=length(genot2$g$index),dimnames=list(NULL,genot))
ppr<-ppl
ppq<-ppl
mconfig<-genot2$mconfig
rmn<-genot2$rmn
if(length(Lfit)>0) {
  if (mconfig["K"]==1){
    pL1<-probs["pL1"]
    pL2<-probs["pL2"]
    gg<-index.genot(cross="B1",M="K",N="M",genot2=genot2)
    ppl[,gg$KKMM]<-rep(c(pL1,(1-pL1)),c(4,4))
    ppl[,gg$KKMm]<-rep(c(pL2,(1-pL2)),c(4,4))
    ppl[,gg$KkMM]<-rev(ppl[,gg$KKMm])
    ppl[,gg$KkMm]<-rev(ppl[,gg$KKMM])
  }
  else{
    pL<-probs["pL"]
    gg<-index.genot(cross="B1",M="M",genot2=genot2)
    ppl[,gg$MM]<-rep(c(pL,(1-pL)),c(4,4))
    ppl[,gg$Mm]<-rev(ppl[,gg$MM])
  }
}
else
  xgen<-c(highL,xgen) #drop LL indices

if(length(Rfit)>0){
  if (mconfig["O"]==1){
    pR1<-probs["pR1"]
    pR2<-probs["pR2"]
    gg<-index.genot(cross="B1",M="N",N="O",genot2=genot2)
    ppr[,gg$NNOO]<-rep(c(pR1,(1-pR1)),4)
    ppr[,gg$NNOo]<-rep(c(pR2,(1-pR2)),4)
    ppr[,gg$NnOO]<-rev(ppr[,gg$NNOo])
    ppr[,gg$NnOo]<-rev(ppr[,gg$NNOO])
  }
  else{
    pR<-probs["pR"]
    gg<-index.genot(cross="B1",N="N",genot2=genot2)

```

```

        ppr[,gg$NN]<-rep(c(pR,(1-pR)),4)
        ppr[,gg$Nn]<-rev(ppr[,gg$NN])
    }
}
else
    xgen<-c(highR,xgen) #drop RR indices

if(length(Qfit)>0) {
    pQ1<-probs["pQ1"]
    pQ2<-probs["pQ2"]
    gg<-index.genot(cross="B1",M="M",N="N",genot2=genot2)
    ppq[,gg$MMNN]<-rep(rep(c(pQ1,(1-pQ1)),c(2,2)),2)
    ppq[,gg$MMNn]<-rep(rep(c(pQ2,(1-pQ2)),c(2,2)),2)
    ppq[,gg$MmNN]<-rev(ppq[,gg$MMNN])
    ppq[,gg$MmNn]<-rev(ppq[,gg$MMNN])
}
else
    xgen<-c(highQ,xgen) #drop QQ indices

w<-ppl*ppq*ppr
dimnames(w)<-list(NULL,genot)
w<-t(w)
#remove QTL genotypes that are not in the model
xgen<-unique(xgen)
if (length(xgen)>0)
    w<-w[,-xgen]
w<-w[markerg,]

w
}
#-----
# weights.f2() : calculate the mixing proportions for a F2 cross
#-----
weights.f2<- function(chosen.model,probs,markerg,genot,genot2,nqgen){

    Qfit<-grep("Q",chosen.model)
    Rfit<-grep("R",chosen.model)
    Lfit<-grep("L",chosen.model)
    mconfig<-genot2$mconfig
    rmn<-genot2$rmn
    rkm<-genot2$rkm
    rno<-genot2$rno

    xgen<-NULL
    qt13<-genot2$qt13
    highL<-grep("LL",qt13)

```

```

hetL<-grep("Ll",qt13)
highR<-grep("RR",qt13)
hetR<-grep("Rr",qt13)
highQ<-grep("QQ",qt13)
hetQ<-grep("Qq",qt13)

ppl<-matrix(1,nrow=27,ncol=length(genot2$g$index),dimnames=list(NULL,genot))
ppr<-ppl
ppq<-ppl

if(length(Lfit)>0) {
  if (mconfig["K"]==1){
    pL1<-probs["pL1"]
    pL2<-probs["pL2"]
    gg<-index.genot(cross="F2",M="K",N="M",genot2=genot2)
    ppl[,gg$KKMM]<-rep(c(pL1^2,2*pL1*(1-pL1),(1-pL1)^2),c(9,9,9))
    ppl[,gg$KKMm]<-rep(c(pL1*pL2,pL2*(1-pL1)+pL1*(1-pL2),(1-pL1)*(1-pL2)),
                      c(9,9,9))
    ppl[,gg$KKmm]<-rep(c(pL2^2,2*pL2*(1-pL2),(1-pL2)^2),c(9,9,9))
    ppl[,gg$KkMM]<-rep(c(pL1*(1-pL2),1-pL1-pL2+ 2*pL1*pL2,pL2*(1-pL1)),
                      c(9,9,9))
    ppl[,gg$KkMm]<-(1/((1-rkm)^2+rkm^2)*
      rep(c((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2),
        (1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2)),
        (1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2)),c(9,9,9)))
    ppl[,gg$Kkmm]<-rev(ppl[,gg$KKMM])
    ppl[,gg$kkMM]<-rev(ppl[,gg$KKmm])
    ppl[,gg$kkMm]<-rev(ppl[,gg$KKMm])
    ppl[,gg$kkmm]<-rev(ppl[,gg$KKMM])
  }
  else{
    pL<-probs["pL"]
    gg<-index.genot(cross="F2",M="M",genot2=genot2)
    ppl[,gg$MM]<-rep(c(pL^2,2*pL*(1-pL),(1-pL)^2),c(9,9,9))
    ppl[,gg$Mm]<-rep(c(pL*(1-pL),pL^2+(1-pL)^2,pL*(1-pL)),c(9,9,9))
    ppl[,gg$mm]<-rev(ppl[,gg$MM])
  }
}
else
  xgen<-c(highL,hetL,xgen) #drop LL,Ll indices

if(length(Rfit)>0){
  if (mconfig["0"]==1){
    pR1<-probs["pR1"]
    pR2<-probs["pR2"]

```

```

gg<-index.genot(cross="F2",M="N",N="0",genot2=genot2)
ppr[,gg$NN00]<-rep(c(pR1^2,2*pR1*(1-pR1),(1-pR1)^2),9)
ppr[,gg$NN0o]<-rep(c(pR1*pR2,pR2*(1-pR1)+pR1*(1-pR2),(1-pR1)*(1-pR2)),9)
ppr[,gg$NNoo]<-rep(c(pR2^2,2*pR2*(1-pR2),(1-pR2)^2),9)
ppr[,gg$Nn00]<-rep(c(pR1*(1-pR2),1-pR1-pR2+ 2*pR1*pR2,pR2*(1-pR1)),9)
ppr[,gg$Nn0o]<-(1/((1-rno)^2+rno^2)*
  rep(c((1-rno)^2*pR1*(1-pR1) + rno^2*pR2*(1-pR2),
    (1-rno)^2*(1-2*pR1*(1-pR1)) + rno^2*(1-2*pR2*(1-pR2)),
    (1-rno)^2*pR1*(1-pR1) + rno^2*pR2*(1-pR2)),9))
ppr[,gg$Nnoo]<-rev(ppr[,gg$Nn00])
ppr[,gg$nn00]<-rev(ppr[,gg$NNoo])
ppr[,gg$nn0o]<-rev(ppr[,gg$NN0o])
ppr[,gg$nnoo]<-rev(ppr[,gg$NN00])
}
else{
  pR<-probs["pR"]
  gg<-index.genot(cross="F2",N="N",genot2=genot2)
  ppr[,gg$NN]<-rep(c(pR^2,2*pR*(1-pR),(1-pR)^2),9)
  ppr[,gg$Nn]<-rep(c(pR*(1-pR),pR^2+(1-pR)^2,pR*(1-pR)),9)
  ppr[,gg$nn]<-rev(ppr[,gg$NN])
}
}
else
  xgen<-c(highR,hetR,xgen) #drop RR,Rr indices

if(length(Qfit)>0) {
  pQ1<-probs["pQ1"]
  pQ2<-probs["pQ2"]
  gg<-index.genot(cross="F2",M="M",N="N",genot2=genot2)
  ppq[,gg$MMNN]<-rep(rep(c(pQ1^2,2*pQ1*(1-pQ1),(1-pQ1)^2),c(3,3,3)),3)
  ppq[,gg$MMNn]<-rep(rep(c(pQ1*pQ2,pQ2*(1-pQ1)+pQ1*(1-pQ2),(1-pQ1)*(1-pQ2)),
    c(3,3,3)),3)
  ppq[,gg$MMnn]<-rep(rep(c(pQ2^2,2*pQ2*(1-pQ2),(1-pQ2)^2),c(3,3,3)),3)
  ppq[,gg$MmNN]<-rep(rep(c(pQ1*(1-pQ2),1-pQ1-pQ2+ 2*pQ1*pQ2,pQ2*(1-pQ1)),
    c(3,3,3)),3)
  ppq[,gg$MmNn]<-(1/((1-rmn)^2+rmn^2)*
    rep(rep(c((1-rmn)^2*pQ1*(1-pQ1) + rmn^2*pQ2*(1-pQ2),
      (1-rmn)^2*(1-2*pQ1*(1-pQ1)) + rmn^2*(1-2*pQ2*(1-pQ2)),
      (1-rmn)^2*pQ1*(1-pQ1) + rmn^2*pQ2*(1-pQ2)),c(3,3,3)),3))
  ppq[,gg$Mmnn]<-rev(ppq[,gg$MmNN])
  ppq[,gg$mmNN]<-rev(ppq[,gg$MMnn])
  ppq[,gg$mmNn]<-rev(ppq[,gg$MMNn])
  ppq[,gg$mmnn]<-rev(ppq[,gg$MMNN])
}
else

```

```

xgen<-c(highQ,hetQ,xgen) #drop QQ indices

w<-ppl*ppq*ppr
dimnames(w)<-list(NULL,genot)
w<-t(w)
#remove QTL genotypes that are not in the model
xgen<-unique(xgen)
if (length(xgen)>0)
  w<-w[,-xgen]
w<-w[markerg,]
w
}

#-----
# mixing.probs() : switching function for calculate the mixing
# proportions depending on the type of breeding design
#-----
mixing.probs<- function(cross,hypothesis,...){
  w<-switch(as.character(cross),
    B1=weights.b1(...),
    B2=weights.b1(...),
    F2=weights.f2(...))
  w
}

#-----
# loglik() : calculate the log-likelihood for an inbred line cross design
# assuming Normal mixture for the trait distribution.
# PARAMETERS of loglik():
# sigma2 - the error variance
# mu.qtl - the component of the mean due to qtl effects
# (a numeric vector whose length is equal to the
# number of qtl genotypes)
# mu.cofactors - the component of the mean due to the
# effects of extra cofactors
# This is a list (grouped by marker type) of numeric vectors
# w - a matrix of mixing weights, whose rows represent marker genotype
# and columns represents qtl genotype.
# w_{ij} is the probability of being in qtl group j given marker i.
# nmgen - the number of marker genotypes
# n - a list containing the sample counts in each marker grouping
# N - the overall sample size
# y - a list of trait values grouped according to marker genotype (ie a
# list of numeric vectors. We assume that, with probability w[i,k],
# y[[i]][j] comes from a Normal distribution with

```

```

#      mean=mu.cofactors[[i]][j]+mu.qtl[k] and variance=sigma2
#-----
loglik<-function(sigma2,mu.qtl,mu.cofactors,w,nmgen,n,N,y){
  lnfj<-function(j,i,w,mu.qtl,mu.cofactors,sigma2,y){
    fy<-y[[i]][j]-mu.cofactors[[i]][j]-mu.qtl
    #fy is a vector of y_{ij}-m_{ij,k} for all k
    fy<-fy*fy/(-2*sigma2)
    fy<-exp(fy)
    sumwf<-w[i,]%*%fy
    #sumwf is a vector of w_{i,k}*f_{i,k}(y_{i,j}) for all k
    log(sumwf)
  }
  lik<-0

  for(i in 1:nmgen){
    likei<-sapply(1:n[i],lnfj,i,w,mu.qtl,mu.cofactors,sigma2,y)
    lik<-lik + sum(likei)
  }

  lik-N/2*log(sigma2*2*pi)
}

#-----
# haldane.pQ1(): calculate pQ1 from pQ2 and rMN if map function is Haldane.
#-----
haldane.pQ1<-function(rmn,pQ2){
  0.5 + 0.5/(1-rmn)*sqrt(1-2*rmn+rmn^2*(1-2*pQ2)^2)
}

#-----
# constrain.b1() : : used in calculating the MLEs of the mixing parameters.
#-----
constrain.b1<-function(pQ1,pQ2,rmn,mapfun){
  #this function carries out an exact maximization

  max.pQ1<-haldane.pQ1(rmn,1-1e-5)

  if (pQ1>max.pQ1)
    pQ1<-max.pQ1
  if (mapfun=="Haldane"){
    cn<-1 #coefficeint of coincidence
    pQ1<-haldane.pQ1(rmn,pQ2)
    if ((pQ1>max.pQ1) && (pQ2<0.5)){
      pQ1<-max.pQ1
    }
  }
}

```



```

        pQ2<-1e-5
    }
    else if (pQ1>max.pQ1){
        pQ1<-max.pQ1
        pQ2<-1-1e-5
    }
    rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
    rQN<-pQ2*rmn/(1-cn*rMQ)
}
else{
    rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
    rQN<-(1-rmn)*(1-pQ1) + rmn*pQ2
    cn<-(1-rmn)*(1-pQ1)/(rMQ*rQN)
}

if (pQ2>1-1e-5){
    pQ2<-1-1e-5
    pQ1<-haldane.pQ1(rmn,pQ2)
    rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
    rQN<-(1-rmn)*(1-pQ1) + rmn*pQ2
}
else if (pQ2<1e-5){
    pQ2<-1e-5
    pQ1<-haldane.pQ1(rmn,pQ2)
    rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
    rQN<-(1-rmn)*(1-pQ1) + rmn*pQ2
}
if (pQ1==0.5) #avioid (1/0) in derivative for infmat
    pQ1<-0.5+1e-5
list(pQ1,pQ2,rMQ,rQN)
}

#-----
# constrain.f2() : used in calculating the MLEs of the mixing parameters.
#-----
constrain.f2<-function(gg,n,en,rmn,mapfun,highQ,hetQ,lowQ,sfrac){
    #Exact maximization is difficult here,
    #we could use numerical maximization algorithm,
    #but in the interest of time,
    #we will just use a pseudo-moment approximation

    n1<-sum(n[gg$MMNN])
    n1Hi<-sum(en[gg$MMNN,highQ])
    n1Lo<-sum(en[gg$MMNN,lowQ])

```

```

p1Hi1<-p1Lo1<-0
if (n1>0){
  p1Hi1<-sqrt(n1Hi/n1)
  p1Lo1<-1-sqrt(n1Lo/n1)
}

n9<-sum(n[gg$mmnn])
n9Hi<-sum(en[gg$mmnn,highQ])
n9Lo<-sum(en[gg$mmnn,lowQ])
p1Lo9<-p1Hi9<-0
if (n9>0){
  p1Lo9<-sqrt(n9Lo/n9)
  p1Hi9<-1-sqrt(n9Hi/n9)
}

pQ1<-((n1Hi*p1Hi1+n1Lo*p1Lo1+
        n9Hi*p1Hi9+n9Lo*p1Lo9)/(n1Hi+n1Lo+n9Hi+n9Lo))

n3<-sum(n[gg$MMnn])
n3Hi<-sum(en[gg$MMnn,highQ])
n3Lo<-sum(en[gg$MMnn,lowQ])
p2Hi3<-p2Lo3<-0
if (n3>0){
  p2Hi3<-sqrt(n3Hi/n3)
  p2Lo3<-1-sqrt(n3Lo/n3)
}

n7<-sum(n[gg$mmNN])
n7Hi<-sum(en[gg$mmNN,highQ])
n7Lo<-sum(en[gg$mmNN,lowQ])
p2Lo7<-p2Hi7<-0
if (n7>0){
  p2Lo7<-sqrt(n7Lo/n7)
  p2Hi7<-1-sqrt(n7Hi/n7)
}

n8<-sum(n[gg$mmNn])
n8Hi<-sum(en[gg$mmNn,highQ])
n8Lo<-sum(en[gg$mmNn,lowQ])

n6<-sum(n[gg$Mmnn])
n6Hi<-sum(en[gg$Mmnn,highQ])
n6Lo<-sum(en[gg$Mmnn,lowQ])

p2Hi6Low8<-p2Low6Hi8<-0
if ((n8>0)&&(n6>0)){
  p2Hi6Low8<-(n6Hi/n6)+(n8Lo/n8)
}

```

```

    p2Low6Hi8<-1-((n6Lo/n6)+(n8Hi/n8))
  }

p2.68<- 0.5*(p2Hi6Low8 +p2Low6Hi8)

n2<-sum(n[gg$MMNn])
n2Hi<-sum(en[gg$MMNn,highQ])
n2Lo<-sum(en[gg$MMNn,lowQ])

n4<-sum(n[gg$MmNN])
n4Hi<-sum(en[gg$MmNN,highQ])
n4Lo<-sum(en[gg$MmNN,lowQ])
p2Hi2Low4<-p2Low2Hi4<-0
if ((n2>0)&&(n4>0)){
  p2Hi2Low4<-(n2Hi/n2)+(n4Lo/n4)
  p2Low2Hi4<-1-((n2Lo/n2)+(n4Hi/n4))
}

p2.24<- 0.5*(p2Hi2Low4 + p2Low2Hi4)

pQ2<-((n3Lo*p2Lo3 + n3Hi*p2Hi3 + n7Lo*p2Lo7 + n7Hi*p2Hi7
      + (n2+n4)*p2.24 + (n6+n8)*p2.68 )
      /(n3Lo+ n3Hi + n7Lo + n7Hi +(n2+n4)+(n6+n8)) )

max.pQ1<-haldane.pQ1(rmn,1-1e-5)
if (pQ1>max.pQ1)
  pQ1<-max.pQ1
if (mapfun=="Haldane"){
  cn<-1 #coefficeint of coincidence
  pQ1<-haldane.pQ1(rmn,pQ2)
  if ((pQ1>max.pQ1) && (pQ2<0.5)){
    pQ1<-max.pQ1
    pQ2<-1e-5
  }
  else if (pQ1>max.pQ1){
    pQ1<-max.pQ1
    pQ2<-1-1e-5
  }
  rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
  rQN<-pQ2*rmn/(1-cn*rMQ)
}
else{
  rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
  rQN<-(1-rmn)*(1-pQ1) + rmn*pQ2
  cn<-(1-rmn)*(1-pQ1)/(rMQ*rQN)
}

```

```

}

if (pQ2>1-1e-5){
  pQ2<-1-1e-5
  pQ1<-haldane.pQ1(rmn,pQ2)
  rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
  rQN<-(1-rmn)*(1-pQ1) + rmn*pQ2
}
else if (pQ2<1e-5){
  pQ2<-1e-5
  pQ1<-haldane.pQ1(rmn,pQ2)
  rMQ<-(1-rmn)*(1-pQ1) + rmn*(1-pQ2)
  rQN<-(1-rmn)*(1-pQ1) + rmn*pQ2
}
if (pQ1==0.5) #aviod (1/0) in derivative for infmat
  pQ1<-0.5+1e-5

list(pQ1,pQ2,rMQ,rQN)
}

#-----
# constrain() : used in calculating the MLEs of the mixing parameters.
#-----
constrain<- function(cross,...){
  w<-switch(as.character(cross),
    B1=constrain.b1(...),
    B2=constrain.b1(...),
    F2=constrain.f2(...))
  w
}

#-----
# phihat.b1(): Calculte MLEs of the mixing parameters for B1.
#-----
phihat.b1<-function(chosen.model,mapfun,n,en,genot,genot2,probs,markerg){
  #sort expected QTL genotype counts by marker category
  cross<-"B1"
  temp<-en
  nqgen<-length(en[,])
  qtl.genotype.labels<-dimnames(en)[[2]]
  en<-matrix(0,nrow=length(genot),ncol=nqgen,dimnames=list(genot,NULL))
  en[markerg,]<-temp[markerg,]
  #sort marker genotype counts by marker category
  n<-n[genot]
  temp<-n

```

```

n<-rep(0,length(genot))
names(n)<-genot
n[markerg]<-temp[markerg]

N<-sum(n)

Qfit<-grep("Q",chosen.model)
Rfit<-grep("R",chosen.model)
Lfit<-grep("L",chosen.model)

highL<-grep("LL",qtl.genotype.labels)
hetL<-grep("Ll",qtl.genotype.labels)
highR<-grep("RR",qtl.genotype.labels)
hetR<-grep("Rr",qtl.genotype.labels)
highQ<-grep("QQ",qtl.genotype.labels)
hetQ<-grep("Qq",qtl.genotype.labels)

probs.hat<-as.list(rep(0,length(probs)))
names(probs.hat)<-names(probs)
mconfig<-genot2$mconfig
rMN<-genot2$rmn
rKM<-genot2$rkM
rNO<-genot2$rno

recomb<-as.list(NULL)
if (length(Qfit)>0){
  gg<-index.genot(cross="B1",M="M",N="N",genot2=genot2)
  denom<-max(1e-5,sum(n[c(gg$MMNN,gg$MmNn)]))
  pQ1.vals<-(1/denom*(sum(en[gg$MMNN,highQ])
    +sum(en[gg$MmNn,hetQ]) ))
  denom<-max(1e-5,sum(n[c(gg$MMNn,gg$MmNN)]))
  pQ2.vals<-(1/denom*(sum(en[gg$MMNn,highQ])
    +sum(en[gg$MmNN,hetQ]) ))
  #rmn.hat<-sum(n[c(gg$MMNn,gg$MmNN)])/sum(n)
  cpQ<-constrain(cross,pQ1.vals,pQ2.vals,rMN,mapfun)
  probs.hat$pQ1<-cpQ[[1]]
  probs.hat$pQ2<-cpQ[[2]]
  recomb$rMQ<-cpQ[[3]]
  recomb$rQN<-cpQ[[4]]
}#end if (length(Qfit)>0)
if (length(Lfit)>0){
  if (mconfig["K"]==1){
    gg<-index.genot(cross="B1",M="K",N="M",genot2=genot2)
    denom<-max(1e-5,sum(n[c(gg$KKMM,gg$KkMm)]))
    pL1.vals<-(1/denom*(sum(en[gg$KKMM,highL])

```

```

        +sum(en[gg$KkMm,hetL]) ))
denom<-max(1e-5,sum(n[c(gg$KKMm,gg$KkMM)]))
pL2.vals<-(1/denom*(sum(en[gg$KKMm,highL])
        +sum(en[gg$KkMM,hetL]) ))
cpL<-constrain(cross,pL1.vals,pL2.vals,rKM,mapfun)
probs.hat$pL1<-cpL[[1]]
probs.hat$pL2<-cpL[[2]]
recomb$rKL<-cpL[[3]]
recomb$rLM<-cpL[[4]]
}
else{
  gg<-index.genot(cross="B1",M="M",genot2=genot2)
  probs.hat$pL<-(1/N*(sum(en[gg$MM,highL])
        +sum(en[gg$Mm,hetL]) ))

  if (probs.hat$pL>1-1e-5)
    probs.hat$pL<- 1-1e-5 #avoid overflows
  else if (probs.hat$pL<0.5+1e-5)
    probs.hat$pL<- 0.5+1e-5
  recomb$rLM<-(1-probs.hat$pL)
}
recomb$rKM<-rKM
}
if (length(Rfit)>0){
  if(mconfig["0"]==1){
    gg<-index.genot(cross="B1",M="N",N="0",genot2=genot2)
    denom<-max(1e-5,sum(n[c(gg$NN00,gg$Nn0o)]))
    pR1.vals<-(1/denom*(sum(en[gg$NN00,highR])
        +sum(en[gg$Nn0o,hetR]) ))
    denom<-max(1e-5,sum(n[c(gg$NN0o,gg$Nn00)]))
    pR2.vals<-(1/denom*(sum(en[gg$NN0o,highR])
        +sum(en[gg$Nn00,hetR]) ))
    cpR<-constrain(cross,pR1.vals,pR2.vals,rN0,mapfun)
    probs.hat$pR1<-cpR[[1]]
    probs.hat$pR2<-cpR[[2]]
    recomb$rNR<-cpR[[3]]
    recomb$rR0<-cpR[[4]]
  }
  else{
    gg<-index.genot(cross="B1",N="N",genot2=genot2)
    probs.hat$pR<-(1/N*(sum(en[gg$NN,highR])
        +sum(en[gg$Nn,hetR]) ))

    if (probs.hat$pR> 1-1e-5)
      probs.hat$pR<- 1-1e-5
    else if (probs.hat$pR<0.5+1e-5)

```

```

        probs.hat$pR<- 0.5+1e-5
        recomb$rNR<-(1-probs.hat$pR)
    }
    recomb$rNO<-rNO
}

recomb$rMN<-rMN
probs.hat<-unlist(probs.hat)
names(probs.hat)<-names(probs)
probs.list<-as.list(1)
probs.list[[1]]<-probs.hat
rec.hat<-unlist(recomb)
names(rec.hat)<-names(recomb)
rec.list<-as.list(1)
rec.list[[1]]<-rec.hat

probs.hat<-list(phi.hat=probs.list,rec.hat=rec.list)
probs.hat
}

#-----
#   phihat.f2(): Calculte MLEs of the mixing parameters for F2.
#-----
phi.hat.f2<-function(chosen.model,mapfun,n,en,genot,genot2,probs,markerg){
  #sort expected QTL genotype counts by marker category
  cross<-"F2"
  temp<-en
  nqgen<-length(en[,1])
  qtl.genotype.labels<-dimnames(en)[[2]]
  en<-matrix(0,nrow=length(genot),ncol=nqgen,dimnames=list(genot,NULL))
  en[markerg,]<-temp[markerg,]
  #sort marker genotype counts by marker category
  n<-n[genot]
  temp<-n
  n<-rep(0,length(genot))
  names(n)<-genot
  n[markerg]<-temp[markerg]
  N<-sum(n)

  Qfit<-grep("Q",chosen.model)
  Rfit<-grep("R",chosen.model)
  Lfit<-grep("L",chosen.model)

  highL<-grep("LL",qtl.genotype.labels)
  hetL<-grep("Ll",qtl.genotype.labels)

```

```

lowL<-grep("ll",qtl.genotype.labels)
highR<-grep("RR",qtl.genotype.labels)
hetR<-grep("Rr",qtl.genotype.labels)
lowR<-grep("rr",qtl.genotype.labels)
highQ<-grep("QQ",qtl.genotype.labels)
hetQ<-grep("Qq",qtl.genotype.labels)
lowQ<-grep("qq",qtl.genotype.labels)

probs.hat<-as.list(rep(0,length(probs)))
names(probs.hat)<-names(probs)
mconfig<-genot2$mconfig
rMN<-genot2$rmn
rKM<-genot2$rkM
rNQ<-genot2$rno

recomb<-as.list(NULL)
if (length(Qfit)>0){
  gg<-index.genot(cross="F2",M="M",N="N",genot2=genot2)
  cpQ<-constrain(cross,gg,n,en,rMN,mapfun,highQ,hetQ,lowQ)
  probs.hat$pQ1<-cpQ[[1]]
  probs.hat$pQ2<-cpQ[[2]]
  recomb$rMQ<-cpQ[[3]]
  recomb$rQN<-cpQ[[4]]
}#end if (length(Qfit)>0)
if (length(Lfit)>0){
  if (mconfig["K"]==1){
    gg<-index.genot(cross="F2",M="K",N="M",genot2=genot2)
    gg2<-list(gg$KKMM,gg$KKMm,gg$KKmm,gg$KkMM,gg$KkMm,gg$Kkmm,gg$kkMM,gg$kkMm,gg$kkmm)
    names(gg2)<-c("MMNN","MMNn","MMnn","MmNN","MmNn","Mmnn","mmNN","mmNn","mmnn")
    cpL<-constrain(cross,gg2,n,en,rKM,mapfun,highL,hetL,lowL)
    probs.hat$pL1<-cpL[[1]]
    probs.hat$pL2<-cpL[[2]]
    recomb$rLM<-cpL[[4]]
  }
  else{
    gg<-index.genot(cross="B1",M="M",genot2=genot2)
    n1<-sum(n[gg$MM])
    n3<-sum(n[gg$mm])
    plHi1<-plLo1<-plLo3<-plHi3<-0
    if (n1>0){
      prlHi1<-sqrt(sum(en[gg$MM,highQ])/n1)
      plLo1<-1-sqrt(sum(en[gg$MM,lowQ])/n1)
    }
    if (n3>0){
      plLo3<-sqrt(sum(en[gg$mm,lowQ])/n3)
    }
  }
}

```



```

        plHi3<-1-sqrt(sum(en[gg$mm,highQ])/n3)
    }
    probs.hat$pL<-0.5*(n1*(plHi1+plLo1)+n3*(plLo3+plHi3))/(n1+n3)

    if (probs.hat$pL>1-1e-5)
        probs.hat$pL<- 1-1e-5
    else if (probs.hat$pL<0.5+1e-5)
        probs.hat$pL<- 0.5+1e-5
    recomb$rLM<-(1-probs.hat$pL)
}
recomb$rKM<-rKM
}

if (length(Rfit)>0){
  if(mconfig["0"]==1){
    gg<-index.genot(cross="F2",M="N",N="0",genot2=genot2)
    gg2<-list(gg$NN00,gg$NN0o,gg$NNoo,gg$Nn00,gg$Nn0o,gg$Nnoo,gg$nn00,gg$nn0o,gg$nnoo)
    names(gg2)<-c("MMNN","MMNn","MMnn","MmNN","MmNn","Mmnn","mmNN","mmNn","mmnn")
    cpR<-constrain(cross,gg2,n,en,rNO,mapfun,highR,heterR,lowR)
    probs.hat$pR1<-cpR[[1]]
    probs.hat$pR2<-cpR[[2]]
    recomb$rNR<-cpR[[3]]
  }
  else{
    gg<-index.genot(cross="F2",N="N",genot2=genot2)

    n1<-sum(n[gg$NN])
    n3<-sum(n[gg$nn])
    prHi1<-prLo1<-prLo3<-prHi3<-0
    if (n1>0){
      prHi1<-sqrt(sum(en[gg$NN,highQ])/n1)
      prLo1<-1-sqrt(sum(en[gg$NN,lowQ])/n1)
    }
    if (n3>0){
      prLo3<-sqrt(sum(en[gg$nn,lowQ])/n3)
      prHi3<-1-sqrt(sum(en[gg$nn,highQ])/n3)
    }
    probs.hat$pR<-0.5*(n1*(prHi1+prLo1)+n3*(prLo3+prHi3))/(n1+n3)

    if (probs.hat$pR>1-1e-5)
        probs.hat$pR<- 1-1e-5
    else if (probs.hat$pR<0.5+1e-5)
        probs.hat$pR<- 0.5+1e-5
    recomb$rNR<-(1-probs.hat$pR)
  }
}

```

```

recomb$rNO<-rNO
}

recomb$rMN<-rMN
probs.hat<-unlist(probs.hat)
names(probs.hat)<-names(probs)
probs.list<-as.list(1)
probs.list[[1]]<-probs.hat
rec.hat<-unlist(recomb)
names(rec.hat)<-names(recomb)
rec.list<-as.list(1)
rec.list[[1]]<-rec.hat

probs.hat<-list(phi.hat=probs.list,rec.hat=rec.list)
probs.hat
}

#-----
# mle.probs() :
# switching function to calculate the MLEs of the mixing parameters
# depending on the type of breeding design
#-----
mle.probs<- function(cross,...){
  probs.hat<-switch(as.character(cross),
    B1=phi.hat.b1(...),
    B2=phi.hat.b1(...),
    F2=phi.hat.f2(...))
  probs.hat
}

#-----
# moment.nw()
#-----
moment.nw <- function(y, mom, sums = F, centered=T,trim=0){
  if (centered==T)
    mu<-mean(y, na.rm = T,trim=trim)
  else mu<-0
  if(sums == F)
    val <- mean((y - mu)^mom, na.rm = T,trim=trim)
  else val <- sum((y - mu)^mom, na.rm = T)
  val
}

#-----
# diff.moments()

```

```

#-----
diff.moments<-function(Y,nmgen,N,n,nc,ind,indc,w,mu.qtl,mu.cofactors,
                        sigma2,vqtl,yvar){
  sig<-sigma2-vqtl

  vq<-mom2.est<-rep(0,nmgen)

  for(i in 1:nmgen) {
    mu.cofaci<-mu.cofactors[[i]]
    num<-length(nc[[i]])
    muq<-wi<-numeric(0)
    for (k in 1:num){
      wi<-c(wi,(nc[[i]][k]/n[i])*w[i,])
      muq<-c(muq,mu.qtl + mean(mu.cofaci[indc[[i]][[k]]]))
    }
    #mom2.est[i]=expected value of var(Yi)
    mom2.est[i]<-(sig+((wi%*%muq^2) - (wi%*%muq)^2))
    nikj<-wi*n[i] #number of each qc in i
    vq[i]<- (nikj/n[i]^2)%*%c(sig/(nikj)+mom2.est[i]/n[i]-2*sig/n[i])

  }

  #vardiff = expected value of var( Ybar_i - Ybar)
  vmarkers<-(n/N)%*%(mom2.est/n+(sig+vqtl)/N -2/N*mom2.est )

  vq<-(n/N)%*%(vq)

  diff<-sig/(vq+vmarkers)

  list(vqtl=vqtl,verr=sig,vqc.within.m=vq, vmarkers=vmarkers,
       totvar=yvar,diff=diff)

}

#-----
# em.known.probs() : calculate (via the EM Algorithm) the maximum
#   likelihood of the observed trait values for inbred linecross
#   data, assuming mixing proportions are not known.
# Descriptions of some of the parameters of em.known.probs()
# Ce - the contrast matrix associated with the QTL genotypes
# MCstar - the coded variables representing the extra marker cofactors
# cofactors.names - the names of the extra cofactors being fitted
# probs0 - a vector of mixing proportions (pL,pR,pQ1,pQ2)
# mapfun - one of "Haldane", "General"
# n - a vector containing the sample counts in each marker grouping

```

```

# nqgen - number of qtl genotypes
# y - a list of trait values grouped according to marker genotype.
# ind,indc,nc,yc - indices, counts and traits for cofactor subgroups
#-----
em.known.probs<-function(chosen.model,cross,hypothesis, Ce,MCstar,
                        cofactors.names,probs,mapfun,rmn,n,nqgen,
                        y,genot,genot2,recomb,nmgen,N,markerg,yvar,ind,indc,nc,yc,
                        calclike=FALSE ,mu.all=FALSE){

  #em Main
  w<-mixing.probs(cross,hypothesis,chosen.model,probs,markerg,genot,
                 genot2,nqgen)
  Z<-matrix(0,nrow=N,ncol=nqgen)
  #eventually Z will store the category identities
  #Y will be a N vector of trait values partitioned by marker group
  Y<-unlist(y) #uppercase Y is a numeric vector, lowercase y is a list
  tCstar.MtM.Cstar<-t(MCstar)%*%MCstar

  for(i in 1:nmgen) #compute e-step for all indivs in group i
    Z[ind[[i]], ]<-cbind(rep(1,n[i])) %*% w[i,]
  X<-cbind(Z)%*%Ce,MCstar) #model matrix

  #m-steps
  eZgivenY.XtX1 <- cbind(t(Ce)%*%diag(c(rep(1,N)%*%Z))%*%Ce,
                        t(Ce)%*%t(Z)%*%MCstar)
  eZgivenY.XtX2 <- cbind(t(MCstar)%*%Z)%*%Ce, tCstar.MtM.Cstar)
  eZgivenY.XtX <- rbind(eZgivenY.XtX1, eZgivenY.XtX2)

  b<- try(solve(eZgivenY.XtX,t(X)%*%Y),silent=TRUE)
  if (inherits(b, "try-error"))
    b<-(ginv(eZgivenY.XtX))%*%t(X)%*%Y
  sigma2<-(1/N)*(t(Y)%*%Y - 2*t(Y)%*%X)%*%b + t(b)%*%eZgivenY.XtX)%*%b)
  sigma2<-as.numeric(sigma2)
  main.names<-dimnames(Ce)[[2]]
  dimnames(b)<-list(c(main.names,cofactors.names),"MLE")
  B<-b[main.names,]
  Bstar<-b[cofactors.names,]
  mu.qtl<-Ce)%*%B
  mu.cofactors<-as.list(n)
  for(i in 1:nmgen)
    mu.cofactors[[i]]<-MCstar[ind[[i]],]%*%Bstar

  model.params<-list(effects=b,variance=sigma2,probs=probs)
  #vqtl<-as.numeric((n/N) %*%((w)%*%mu.qtl^2) - (w)%*%mu.qtl^2))
  #the above is equal to
  vqtl<- as.numeric((1/N)*( t(b)%*%( eZgivenY.XtX-t(X)%*%X)%*%b))

```

```

mvars<- diff.moments(Y,nmgen,N,n,nc,ind,indc,w,mu.qtl,mu.cofactors,
                    sigma2,vqtl,yvar)

if (calclike==TRUE){
  newlike<-loglik(sigma2,mu.qtl,mu.cofactors,w,nmgen,n,N,y)
  val<-list(model.params=model.params,recomb=recomb,
            loglike=newlike,mvars=mvars)
}
else val<-list(model.params=model.params,recomb=recomb,mvars=mvars)

if (mu.all==TRUE){
  val$mu.qtl <-mu.qtl
  val$mu.cofactors <-mu.cofactors
}
val
}

#-----
#  getZij(): for the e-step: estimate the category identity of each indiv in group i
#-----

getZij<-function(j,i,w,mu.qtl,mu.cofactors,sigma2,y){
  #e-step: estimate the category identity of each indiv in group i
  fy<-y[[i]][j]-mu.cofactors[[i]][j]-mu.qtl
  #fy is a vector of  $y_{ij}-m_{ij,k}$  for all k
  fy<-fy*fy/(-2*sigma2)
  fy<-exp(fy)
  wf<-w[i,]*fy #vector of  $w_{i,k}*f_{i,k}(y_{i,j})$  for all k
  wf/sum(wf) #vector of  $z_{ijk}$  for a specific (i,j) pair and for all k
}

#-----
# em.unknown.probs() : calculate (via the EM Algorithm)
#   the maximum likelihood of the observed trait values
#   for inbred linecross data, where the mixing proportions are unknown.
#
# Descriptions of some of the parameters of em.unknown.probs():
# Ce - the contrast matrix associated with the QTL genotypes
# MCstar - the coded variables representing the extra cofactors
# cofactors.names - the names of the extra cofactors being fitted
# sigma20 -initial value for the variance.
# b0 - initial values for the effects of all factors being fitted
#   (intercept, qtl effects, and effects of any extra cofactors)
# probs0 - initial mixing proportions, a numeric vector
# tol - the MLE is found when
# maxit - the maximum number of iterations allowed.

```

```

# mapfun - one of "Haldane", "General"
# n - a list containing the sample counts in each marker grouping
# nqgen - number of qtl genotypes
# y - a list of trait values grouped according to marker genotype.
# ind,indc,nc,yc - indices, counts and traits for cofactor subgroups
#-----
em.unknown.probs<-function(chosen.model,cross,hypothesis,Ce,MCstar,
                           cofactors.names,sigma20, b0,probs0,tol,maxit,
                           mapfun,n,nqgen,y,genot,genot2,startvals=FALSE,recomb,
                           imat.type="expected",mvars,ind,indc,nc,yc){

  #em Main
  rmn<-genot2$rmn
  rkm<-genot2$rkm
  rno<-genot2$rno

  nmgen<-length(n) #number of marker geneotypes
  markerg<-names(y) #names of the marker genotypes
  N<-sum(n)
  sigma2<-sigma20
  probs<-probs0 #the mixing proportions

  main.names<-dimnames(Ce)[[2]]

  b<-b0
  dimnames(b)<-list(c(main.names,cofactors.names),"MLE")
  B<-b[main.names,]
  Bstar<-b[cofactors.names,]

  Z<-matrix(0,nrow=N,ncol=nqgen)
  #eventually Z will store the category identities

  mu.qtl<-Ce%*%B
  mu.cofactors<-as.list(n)
  for(i in 1:nmgen)
    mu.cofactors[[i]]<-MCstar[ind[[i]],]%*%Bstar
  w<-mixing.probs(cross,hypothesis,chosen.model,probs,markerg,genot,
                  genot2,nqgen)

  rMQ.config<-"Not Used" #will hold informational stuff
  count<-0
  stopcond<-tol+1
  oldlike<-loglik(sigma2,mu.qtl,mu.cofactors,w,nmgen,n,N,y)
  startlike<-oldlike
  newlike<-oldlike

```

```

#Y will be a N vector of trait values partitioned by marker group
Y<-unlist(y) #uppercase Y is a numeric vector, lowercase y is a list
tCstar.MtM.Cstar<-t(MCstar)%*%MCstar
hardstop<-F
yvar<-moment.nw(Y,2)

while((stopcond>tol) && (count<maxit)&&(!hardstop) ){
  prev<-list(b=b,sigma2=sigma2,probs=probs,w=w,
    mu.qtl=mu.qtl,mu.cofactors=mu.cofactors,
    newlike=newlike)
  #e-step
  for(i in 1:nmgen) #compute e-step for all indivs in group i
    Z[ind[[i]], ]<-t(sapply(1:n[i],getZij,i,w,mu.qtl,mu.cofactors,
      sigma2,y))
  X<-cbind(Z%*%Ce,MCstar) #model matrix

  #m-steps
  eZgivenY.XtX1<-cbind(t(Ce)%*%diag(c(rep(1,N)%*%Z))%*%Ce,
    t(Ce)%*%t(Z)%*%MCstar)
  eZgivenY.XtX2<-cbind(t(MCstar)%*%Z%*%Ce, tCstar.MtM.Cstar)
  eZgivenY.XtX<-rbind(eZgivenY.XtX1, eZgivenY.XtX2)

  b<- try(solve(eZgivenY.XtX,t(X)%*%Y),silent=TRUE)
  if (inherits(b, "try-error"))
    b<-(ginv(eZgivenY.XtX))%*%t(X)%*%Y
  sigma2<-(1/N)*(t(Y)%*%Y - 2*t(Y)%*%X%*%b + t(b)%*%eZgivenY.XtX%*%b)
  sigma2<-as.numeric(sigma2)

  dimnames(b)<-list(c(main.names,cofactors.names),"MLE")
  B<-b[main.names,]
  Bstar<-b[cofactors.names,]
  mu.qtl<-Ce%*%B
  mu.cofactors<-as.list(n)
  for(i in 1:nmgen)
    mu.cofactors[[i]]<-MCstar[ind[[i]],]%*%Bstar

  #calculate the expected QTL genotype counts and get MLE of probs
  qtl.labels<-dimnames(Ce)[[1]]
  en<-matrix(0,nrow=nmgen,ncol=nqgen,dimnames=list(names(n),qtl.labels))
  for(i in 1:nmgen) #en_{ij}=expected number of indivs in group ij
    en[i, ]<- rep(1,n[i])%*%rbind(Z[ind[[i]], ])
  best<-probs
  rbest<-recomb
  wbest<-w

```

```

probs<-mle.probs(cross,chosen.model,mapfun,n,en,genot,genot2,
                probs,markerg)
probs.list<-probs$phi.hat
rec.list<-probs$rec.hat
#print(probs.list)
bestlike<-oldlike
improved<-F

for (i in 1:length(probs.list)){
  probs<-probs.list[[i]]
  recc<-rec.list[[i]]

  w<-mixing.probs(cross,hypothesis,chosen.model,probs,markerg,
                  genot,genot2,nqgen)
  newlike<-loglik(sigma2,mu.qtl,mu.cofactors,w,nmgen,n,N,y)
  vqtl<- as.numeric((1/N)*( t(b)%*%( eZgivenY.XtX-t(X)%*%X)%*%b))
  mvars2<- diff.moments(Y,nmgen,N,n,nc,ind,indc,w,mu.qtl,
                        mu.cofactors,sigma2,vqtl,yvar)
  model.params.hat<-list(effects=b,variance=sigma2,probs=probs)

  if (newlike>=bestlike){
    mvars<-mvars2
    improved<-T
    best<-probs
    rbest<-recc
    wbest<-w
    bestlike<-newlike
    qtlvar<-vqtl
  }
}
probs<-best[names(probs)]
recomb<-rbest
w<-wbest
newlike<-bestlike
if (!improved){
  hardstop<-T
  #in the unlikely event that this happens, rollback
  b<-prev$b
  sigma2<-prev$sigma2
  probs<-prev$probs
  newlike<-prev$newlike
  w<-prev$w
  mu.cofactors<-prev$mu.cofactors
  mu.qtl<-prev$mu.qtl
  for(i in 1:nmgen)

```



```

        Z[ind[[i]], ]<-t(sapply(1:n[i],getZij,i,w,
                                mu.qtl,mu.cofactors,sigma2,y))
    }

    count<-count+1
    stopcond<-abs(oldlike-newlike)
    oldlike<-newlike
} #end while

#format the output
if ((mapfun=="Haldane")&&(startvals==FALSE)) {
    hprobs<-haldane.probs(chosen.model,probs,recomb)
    probs<-hprobs$probs
    recomb<-hprobs$recomb
}

conv.info<-c(tol,stopcond,count)
names(conv.info)<-c("chosen.tolerance","actual.tolerance","num.iterations")
model.params<-list(effects=b,variance=sigma2,probs=probs)

if(!startvals){
    if (imat.type=="expected")
        test<-emcov.fisher(chosen.model,cross,hypothesis,mapfun,genot,
                            genot2,markerg,model.params,MCstar,Ce,cofactors.names,
                            n,nqgen,nmgen,recomb)
    else
        test<-emcov.observed(chosen.model,cross,hypothesis,mapfun,genot,
                              genot2,markerg,model.params,Z,MCstar,Ce,cofactors.names,
                              n,nqgen,nmgen,y,recomb)

    model.params$effects<-cbind(model.params$effects,test$b.result)
    model.params$probs<-cbind(model.params$probs,test$probs.result)
    dimnames(model.params$probs)[[2]][1]<-"MLE"
    names(model.params$variance)<-"MLE"

    val<-list(convergence.info=conv.info,model.params=model.params,
              infmat.is.singular=test$infmat.singular,
              recomb=recomb,loglike=newlike, startlike=startlike,
              hardstop=hardstop,mvars=mvars)
}
else{
    val<-list(model.params=model.params,recomb=recomb,loglike=newlike,
              hardstop=hardstop,mvars=mvars)
}
val

```

```

}

#-----
# haldane.probs()
#-----

haldane.probs<-function(chosen.model,probs,recomb){
  Lfit<-grep("L",chosen.model)
  Qfit<-grep("Q",chosen.model)
  Rfit<-grep("R",chosen.model)
  if (length(Qfit)>0){
    pQ1.index<-pmatch("pQ1",names(probs))
    if (!is.na(pQ1.index)){
      recomb["pQ1"]<-probs["pQ1"]
      probs<-probs[-pQ1.index]
    }
  }
  if (length(Lfit)>0){
    pL1.index<-pmatch("pL1",names(probs))
    if (!is.na(pL1.index)){
      recomb["pL1"]<-probs["pL1"]
      probs<-probs[-pL1.index]
    }
  }
  if (length(Rfit)>0){
    pR1.index<-pmatch("pR1",names(probs))
    if (!is.na(pR1.index)){
      recomb["pR1"]<-probs["pR1"]
      probs<-probs[-pR1.index]
    }
  }
  list(probs=probs,recomb=recomb)
}

#-----
# cim.H0.regress() :
#   calculate, via linear regression, the maximum likelihood
#   of the observed trait values for inbred linecross data, for H0 of no QTL anywhere
#-----

cim.H0.regress<-function(MCstar,cofactors.names,n,y,genot,genot2,startvals){
  #em Main
  nmgen<-length(n) #nmgen=number of marker geneotypes
  markerg<-names(y) #names of the marker genotypes
  N<-sum(n)

  Z<-matrix(1,nrow=N,ncol=1)

```

```

#Z only stores coded values for the intercept
#We need an index to identify the marker groups
ind<-vector("list", nmgen)
names(ind)<-names(n)
ind[[1]]<-1:n[1]
for(i in 2:nmgen)
  ind[[i]]<- (1+sum(n[1:(i-1)])):sum(n[1:i])

#Y will be a N vector of trait values partitioned by marker group
Y<-unlist(y) #uppercase Y is a numeric vector, lowercase y is a list

X<-cbind(Z,MCstar)      #model matrix
#m-steps
XtX <- t(X)%*%X
b<- try(solve(XtX,t(X)%*%Y),silent=TRUE)
if (inherits(b, "try-error"))
  b<-(ginv(XtX))%*%t(X)%*%Y
sigma2<-(1/N)*(t(Y)%*%Y - 2*t(Y)%*%X%*%b + t(b)%*%XtX%*%b)
sigma2<-as.numeric(sigma2)
dimnames(b)<-list(c("(Intercept)",cofactors.names),"MLE")
B<-b["(Intercept)",]
Bstar<-b[cofactors.names,]
mu.qtl<-B
mu.cofactors<-as.list(n)
  for(i in 1:nmgen)
    mu.cofactors[[i]]<-MCstar[ind[[i]],]%*%Bstar
w<-matrix(1,nmgen,1,dimnames=list(marker, NULL))

#record value of loglikelihood of observed data based on updated mles.
newlike<-loglik(sigma2,mu.qtl,mu.cofactors,w,nmgen,n,N,y)

model.params<-list(effects=b,variance=sigma2)
recomb<-c(rMN=genot2$rmn)

if(!startvals){
  #calculatr information matrix
  imat<-XtX
  eigen.values.imat<-eigen(imat,TRUE,TRUE)$values
  imat.is.singular<-any(eigen.values.imat<=.Machine$double.eps)
  infmat.singular<-FALSE
  if(imat.is.singular){
    infmat.singular<-TRUE
    imat.inv<-ginv(imat)
  }
  else

```

```

        imat.inv<-solve(imat)
        imat.inv<-imat.inv*sigma2
        var.all<-diag(imat.inv)
        std.err.beta<-sqrt(var.all)
        zstat0.beta<-model.params$effects[,1]/std.err.beta
        pval.beta<-2*(1-pnorm(abs(zstat0.beta)))
        b.result<-cbind(std.err.beta,zstat0.beta,pval.beta)
        dimnames(b.result)<-list(dimnames(model.params$effects)[[1]],
                                c("std.err", "z0", "P>|z0|"))
        model.params$effects<-cbind(model.params$effects,b.result)
        names(model.params$variance)<- "MLE"

        val<-list(model.params=model.params,
                  infmat.is.singular=infmat.singular,recomb=recomb,loglike=newlike)
    }

    else
        val<-list(model.params=model.params,recomb=recomb,loglike=newlike)

    val
}

```

B.2 Utility functions for QTL analysis (R Code)

Table B.2: List of utility functions

Function	Description
lm.linecross()	Carries out Linear regression and stepwise regression for F2 and Backcross data (requires functions contrasts.b1(), contrasts.f2() given in Section B.1).
d.binomial() d.felsenstein() d.haldane() d.kosambi() d.morgan()	Genetic Map functions
r.binomial() r.felsenstein() r.haldane() r.kosambi() r.morgan()	Inverse Genetic Map functions
recomb.matrix()	Calculates the distance matrix from recombination fraction of adjacent markers.
cro.import() rqtl.import() zmapqtl.import()	These functions import <i>QTL cartographer</i> files into to R/S-PLUS objects. cro.import(): import a Rcross input file of the form: 'cross.inp' rqtl.import(): import a Rqtl output file of the form: 'Rqtl.out' zmapqtl.import(): import a Zmapqtl output file: 'Zmapqtl.out'

Source Code

```
#-----
# lm.linecross() : carries out Linear regression and stepwise regression for F2 and Backcross data
#-----
lm.linecross <- function(cross,data, regressors, all.markers,markers.to.fit,
                        homog.high, heteroz, homog.low, trait, step = F, trace = 1){
  m <- match.call()
  x<-names(data[,regressors])
  marker.id<- pmatch(x,all.markers)
  if (is.null(markers.to.fit))
    flanking.markers<-x
```

```

else flanking.markers<-markers.to.fit
marker.design.frame<-data.frame(data[!is.na(data[, trait]),flanking.markers])
marker.contrasts<-switch(as.character(cross),
  B1=lapply(marker.design.frame,contrasts.b1,AA=homog.high, Aa=heteroz,hi="AA"),
  B2=lapply(marker.design.frame,contrasts.b2,Aa=heteroz,aa=homog.low,hi="aa"),
  F2=lapply(marker.design.frame,contrasts.f2,AA=homog.high, Aa=heteroz,aa=homog.low,hi="AA")
)
sum.flank<-paste(flanking.markers,collapse="+")
cofactors.formula<-formula(paste(trait,"~",sum.flank))
args <- list(formula = cofactors.formula, data = m$data,
  na.action = as.name("na.omit"),
  contrasts = as.name("marker.contrasts"), singular.ok = T)
genotypic.mean.lm <- do.call("lm", args)
if(step == T) {
  assign("contr", genotypic.mean.lm$contrasts, 1)
  low <- paste("~", paste(x, collapse = " + "))
  upp <- paste("~", paste(all.markers, collapse = " + "))
  genotypic.mean.lm <- suppressWarnings( stepAIC(genotypic.mean.lm,
    scope = list(upper = upp, lower = low), trace = trace))
}
genotypic.mean.lm
}
# EXAMPLE of using lm.linecross()
y3<-lm.linecross("B1",b1s1$data, 21:22, b1s1$markers[15:25],
  b1s1$markers[15:25],"AA", "Aa","aa", "t1", step = T, trace = 1)
print(y3)
#what markers were selected at the end of the stepwise selection?
markers<- attr(y3$terms,"predvars")[-1]
print(markers)
#or
markers<- attr(y3$terms,"term.labels")
print(markers)

#-----
# map functions: d.binomial(), d.felsenstein(), d.haldane(), d.kosambi(), d.morgan()
#-----
d.binomial <- function(r, N){
  #returns distance in Morgans
  0.5 * N * (1 - (1 - 2 * r)^(1/N))
}
d.felsenstein <- function(r, k){
  #returns distance in Morgans
  1/(2 * (k - 2)) * log((1 - 2 * r)/(1 - 2 * (k - 2) * r))
}
d.haldane <- function(r){

```

```

    #returns distance in Morgans
    ifelse(r < 0.5, -0.5 * log(1 - 2 * r), Inf)
}
d.kosambi <- function(r){
    #returns distance in Morgans
    0.5 * atanh(2 * r)
}
d.morgan <- function(r){
    #returns distance in Morgans
    r
}

#-----
# inverse map functions: r.binomial(), r.felsenstein(), r.haldane(), r.kosambi(), r.morgan()
#-----

r.binomial <- function(d, N){
    #d=distance in Morgans
    #N is the max no of crossovers assumed
    ifelse(d < N/2, 0.5 * (1 - (1 - (2 * d)/N)^N), 0.5)
}
r.felsenstein <- function(d, k){
    #d=distance in Morgans
    (1 - exp(2 * (k - 2) * d))/(2*(1 - (k - 1) * exp(2*(k - 2) * d)))
}
r.haldane <- function(d){
    #d=distance in Morgans
    0.5 * (1 - exp(-2 * abs(d)))
}
r.kosambi <-function(d){
    #d=distance in Morgans
    0.5 * tanh(2 * d)
}
r.morgan <- function(d){
    #d=distance in Morgans
    d
}

#-----
# recomb.matrix() : function for calculating the distance matrix from
# recombination frequency of adjacent markers.
#-----

recomb.matrix <- function(r.curr.next = NULL, d.curr.next = NULL,
    Units = "Morgans", return.val = "recomb", mapfun = "Haldane", ...){
    #r.curr.next is a vector and d.curr.next is a vector
    #return.val may be "recomb"

```

```

#(to return a matrix of recombination fractions)
# or "distance" (to return a matrix of map distances in Morgans)
#Units may be "Morgans" or "cM" (the units of d.curr.next)
#mapfun may be "Haldane","Kosambi","Morgan","Felsenstein","Binomial"
m <- match.call()
if(is.null(r.curr.next) && is.null(d.curr.next))
  stop(paste("neither recombination fractions nor distances",
             "between consecutive loci were supplied"))
if(!((Units == "Morgans") || (Units == "cM")))
  stop(paste("Units must be 'Morgans' or 'cM', found",
             m$units))
if(!((return.val == "recomb") || (return.val == "distance")))
  stop(paste("return.val must be 'recomb' or 'distance', found",
             m$return.val))
mapfuncs <- switch(as.character(mapfun),
  Haldane = T,
  Kosambi = T,
  Morgan = T,
  Felsenstein = T,
  Binomial = T,
  F)
if(!mapfuncs)
  stop(paste("mapfun must be one of 'Haldane', 'Kosambi',",
             "'Morgan', 'Felsenstein', 'Binomial'"))
if(!is.null(r.curr.next)) {
  if(any(r.curr.next > 0.5))
    stop("invalid recombination frequencies")
  if(length(r.curr.next) == 1.)
    r.curr.next <- c(r.curr.next, 0.)
  d.curr.next <- switch(as.character(mapfun),
    Haldane = d.haldane(r.curr.next),
    Kosambi = d.kosambi(r.curr.next),
    Morgan = r.curr.next,
    Felsenstein = d.felsenstein(r.curr.next, k),
    Binomial = d.binomial(r.curr.next, N))
}
else {
  if(length(d.curr.next) == 1.)
    d.curr.next <- c(d.curr.next, 0.)
  #convert to Morgans
  if(Units == "cM") d.curr.next <- 0.01 * d.curr.next
}
x <- d.curr.next
lenx <- length(x)
x[2:lenx] <- x[ - lenx]

```



```

x[1.] <- 0.
dmat <- matrix(x, lenx, lenx)
dimnames(dmat) <- list(names(x), names(x))
dmat[row(dmat) <= col(dmat)] <- 0.
dmat <- apply(dmat, 2., cumsum)
dmat <- dmat + t(dmat)
rmat <- switch(as.character(mapfun),
  Haldane = apply(dmat, 2., r.haldane),
  Kosambi = apply(dmat, 2., r.kosambi),
  Morgan = dmat,
  Felsenstein = apply(dmat, 2., r.felsenstein, k),
  Binomial = apply(dmat, 2., r.binomial, N))
dimnames(rmat) <- list(names(x), names(x))
switch(as.character(return.val),
  distance = dmat,
  rmat)
}

#-----
# functions for importing QTL cartographer files into to R/S-PLUS objects.
# cro.import(): import a Rcross input file of the form: 'cross.inp'
# rqtl.import(): import a Rqtl output file of the form: 'Rqtl.out'
# zmapqtl.import(): import a Zmapqtl output file: 'Zmapqtl.out'
#-----
cro.import <- function(filename){
  cat("\n", file = filename, append = T)
  #ensure newline before eof for S-PLUS6
  yn <- scan(filename, "")
  if(!(yn[4] == "cross.inp")){
    print(yn[4])
    stop("Invalid Rcross input file: 'cross.inp' not found")
  }
  cross.index <- grep("-Cross", yn)
  traits.index <- grep("-traits", yn)
  otraits.index <- grep("-otraits", yn)
  sampsize.index <- grep("-SampleSize", yn)
  case.index <- grep("-case", yn)
  if(length(cross.index) == 0)
    stop("missing '-Cross' flag")
  if(length(traits.index) == 0)
    stop("missing '-traits' flag")
  if(length(sampsize.index) == 0)
    stop("missing '-SampleSize' flag")
  if(length(otraits.index) == 0)

```

```

    stop("missing '-otraits' flag")
  if(length(case.index) == 0)
    stop("missing '-case' flag")
  cross <- yn[cross.index + 1]
  traits <- as.numeric(yn[traits.index + 1])
  missing.index <- grep("-missingtrait", yn)
  if(length(missing.index) > 0) {
    missing.trait <- yn[grep("-missingtrait", yn) + 1]
    names(missing.trait) <- missing.index
  }
  else missing.trait <- NA
  otrait <- as.numeric(yn[otraits.index + 1])
  sampsize <- as.numeric(yn[sampsize.index[1] + 1])
  case <- yn[case.index + 1]
  starts <- grep("-start", yn)
  if(length(starts) > 0) {
    starts.what <- yn[starts + 1]
    names(starts) <- starts.what
    if((sampsize.index == 0) || (sampsize == 0))
      stop("sample size should be greater than zero")
    by.indiv <- length(starts.what[starts.what == "individuals"]) > 0
    by.column <- length(starts.what[(starts.what == "markers" |
      (starts.what == "traits" | (starts.what == "otraits"))]) > 0
    ok.by.column <- ( (length(starts.what[(starts.what == "markers")]) >= 1)
      && (length(starts.what[(starts.what == "traits")]) >= 1))
    if(by.indiv && by.column)
      stop("Data may be 'by individuals (row)' or 'by column', not both.")
    if(by.column && (!ok.by.column))
      stop("file should contain at least one marker and at least one trait")
  }
  else stop("found no -start flags")
  stops <- grep("-stop", yn)
  if(length(stops) > 0) {
    stops.what <- yn[stops + 1]
    names(stops) <- stops.what
  }
  else stop("found no -stop flags")
  if(length(starts) != length(stops))
    stop(paste( "Invalid QTL Cartographer, Rcross input file:",
      "unequal numbers of '-start' and '-stop' flags"))
  test1 <- (stops.what == starts.what)
  if(length(test1[test1 == T]) != length(stops.what))
    stop(paste("Invalid QTL Cartographer, Rcross input file:",
      "'-start' and '-stop' flags do not match"))
  if(by.indiv)

```

```

stop("input in the 'by individuals' format not supported by cro.import.")

if(ok.by.column) {
  markerind <- cbind(starts[names(starts) == "markers"] + 2,
    stops[names(stops) == "markers"] - 1)
  index.markers <- apply(markerind, 1, FUN = function(h){h[1]:h[2]})
  index.markers <- unlist(index.markers)
  transtab <- grep("-TranslationTable", yn)
  translation.table <- t(matrix(yn[(transtab + 1):(transtab + 18)],
    nrow = 3, ncol = 6))
  marker.data <- matrix(yn[index.markers], nrow = sampsize + 1)
  markers <- make.names(marker.data[1, ])
  marker.data <- cbind(marker.data[-1, ])
  dimnames(marker.data) <- list(NULL, markers)

  traitind <- cbind(starts[names(starts) == "traits"] + 2,
    stops[names(stops) == "traits"] - 1)
  index.traits <- apply(traitind, 1, FUN = function(h){h[1]:h[2]})
  index.traits <- unlist(index.traits)
  trait.data <- matrix(yn[index.traits], nrow = sampsize + 1)
  traits <- make.names(trait.data[1, ])
  trait.data <- cbind(trait.data[-1, ])
  misstrait <- (trait.data == missing.trait[1])
  trait.data[misstrait] <- NA
  trait.data <- apply(trait.data, 2, as.numeric)
  dimnames(trait.data) <- list(NULL, traits)

  if(otraits > 0) {
    otraitind <- cbind(starts[names(starts) == "otraits"] + 2,
      stops[names(stops) == "otraits"] - 1)
    index.otraits <- apply(otraitind, 1, FUN = function(h){h[1]:h[2]})
    index.otraits <- unlist(index.otraits)
    otrait.data <- matrix(yn[index.otraits], nrow = sampsize + 1)
    otrait <- make.names(otrait.data[1, ])
    otrait.data <- cbind(otrait.data[-1, ])
    missotraits <- (otrait.data == missing.trait[1])
    otrait.data[missotraits] <- NA
    otrait.data <- apply(otrait.data, 2, as.numeric)
    dimnames(otrait.data) <- list(NULL, otrait)
    dat <- data.frame(marker.data, trait.data, otrait.data)
    cartdata <- list(data = dat, identifier = yn[2], cross = cross,
      , format = yn[4], transtab = translation.table, markers =
        markers, traits = traits, otrait = otrait)
  }
}
else {

```

```

        dat <- data.frame(marker.data, trait.data)
        cartdata <- list(data = dat, identifier = yn[2], format = yn[4],
            cross = cross, transtab = translation.table, markers
            = markers, traits = traits)
    }
}
cartdata
}

#-----
rqtl.import <- function(filename){
  cat("\n", file = filename, append = T)
  #ensure newline before eof for splus6
  yn <- scan(filename, "")
  if(!(yn[4] == "Rqtl.out"))
    stop("Invalid Rqtl output file: 'Rqtl.out' not found")
  ntraits.index <- grep("-t", yn)
  nqtls.index <- grep("-k", yn)
  qtldat.index <- grep("-l", yn)
  if(length(ntraits.index) == 0)
    stop("missing '-l' flag")
  if(length(nqtls.index) == 0)
    stop("missing '-k' flag")
  if(length(qtldat.index) == 0)
    stop("missing '-l' flag, no QTL info found.")
  ntraits <- as.numeric(yn[ntraits.index[1] + 1])
  nqtl <- as.numeric(yn[nqtls.index[1] + 1])
  first.index.qtl <- qtldat.index[2]
  last.index.qtl <- qtldat.index[nqtl + 1] + 7
  index.qtls <- first.index.qtl:last.index.qtl
  qtl.data <- matrix(yn[index.qtls], nrow = 8)
  qtl.data <- qtl.data[-1, ]
  if(nqtl == 1)
    qtl.data <- rbind(as.numeric(qtl.data))
  else qtl.data <- apply(qtl.data, 1, as.numeric)
  qtl.data <- data.frame(qtl.data)
  qtldat <- cbind(qtl.data[, 1:3], qtl.data[, 3] + 1, qtl.data[, 4:7])
  datnames <- c("qtl.Q", "chromosome", "marker.M", "marker.N", "recomb.MQ",
    "recomb.QN", "additive.Q", "dominance.Q")
  names(qtldat) <- datnames
  data.frame(qtldat)
}

#-----
zmapqtl.import<-function(filename,type="backcross"){
  #ensure newline before eof for S-PLUS6
  cat("\n",file=filename,append=T)

```

```

yn<-scan(filename,"")
if (!(yn[4]=="Zmapqtl.out")) stop("Invalid file: 'Zmapqtl.out' not found")

output.start<-grep("^-s",yn)
output.end<-grep("^-e",yn)

if (length(output.start)==0) stop("missing '-s' flag")
if (length(output.end)==0)
  stop("missing '-e' flag")
index.data<-(output.start+1):(output.end-1)
zmap.names<-yn[(output.start-21):(output.start-1)]
zmap.names<- make.names(zmap.names)
zmap.data<-matrix(yn[index.data],nrow=21)
zmap.data<-apply(zmap.data,1, as.numeric)
dimnames(zmap.data)[[2]]<-zmap.names
if (type=="backcross")
  zmap.data<-zmap.data[,1:8]
zmap.data
}

```

B.3 Examples of using the utility functions with QTL Cartographer

```
#-----
# Using the QTL Cartographer module Rmap, generate the marker map
#       Simulate a map of 2 chromosomes-random number of
#       markers on each, equally spaced, 10cM apart
#
# QTL Cartographer is copyright:
# Copyright(C) 1194-2001 C. J. Batsen, B. S. Weir and Z. B. Zeng
#-----

remove(list=ls())
wkdir<-"/u/students/nwill/newsims/"
#if a QTL Cartographer resource file exists here, then remove it.
system("rm qtlcart.rc")
#get a random seed to submit to QTL Cartographer
set.seed(120)
myseed<-sample(1e6, 1)
rmap.call<-paste("Rmap -A -V -W", wkdir, "-s", myseed,
  "-e b1simmap.log -o b1sim.map -g 3 -f 1 -c 2 -m 20 -vm 4.0",
  "-d 10 -t 0.0")
k<-system(rmap.call) #R,
#note: call requires path to QTL Cartographer to be in your PATH variable
#k<-unix(rmap.call, output=T) #S-PLUS
#k<-dos(rmap.call, output=T, trans=F) #S-PLUS

#-----
# Here is an example of using the QTL Cartographer module Rqtl to define QTL positions and effects.
# Here we sprinke nine (9) qtl onto the map.
#-----

myseed<-57627453
rqtl.call<-paste("Rqtl -A -V -W", wkdir, "-s", myseed,
  "-e b1simqtl.log -o b1sim.qtl -m b1sim.map -t 1 -q 9 -d 4",
  "-b 2.0 -1 2.0 -2 2.0 -E 0.0")
k<-system(rqtl.call) #R
#k<-unix(rqtl.call, output=T) #S-PLUS
#k<-dos(rqtl.call, output=T, trans=F) #S-PLUS

#note: to create more variety, a different 'b1sim.qtl' file
#(different from the one generated by the above code) was used for our simulations.
#maps haveing 11 Qtl and one Qtl respectively, were used instead, and they
#were arbitrarily chosen by hand, they were not generated by Rqtl.
```

```

#-----
#   Using the QTL Cartographer module Rcross, generate the samples... nsamp of them
#-----

nsamp=100 #the number of replicate samples
sampsiz=2000 #the sample size
set.seed(153) #initialize for reproducibility
for (i in 1:nsamp){
  b1datafile<-paste("b1s",i,".cro1", sep="")
  myseed<-sample(5e7:6e7, 1) #different seed each time
  rcross.call<-paste("Rcross -A -V -W",wkdir,"-s", myseed,"-n",
    sampsiz,"-o",b1datafile, "-e b1sim.log -m b1sim.map",
    "-q b1sim.qtl -g 1 -c B1 -H 0.5 -I 0")
  k<-system(rcross.call)
}

#-----
#   Using the utility function cro.import(),import the samples into R/S-PLUS objects
#-----

source("cro.import.r")
for (i in 1:nsamp){
  b1datafile<-paste("b1s",i,".cro1", sep="")
  b1object<-paste("b1s",i, sep="")
  assign(b1object, cro.import(paste(wkdir,b1datafile,sep="")))
}

#-----
# Using the utility function rqt1.import() to
# import the marker map and qtl information in order to plot the map
#-----

source("vuwfunc.r")
source("cro.import.r")
b1sim100.chrom1<-read.table(paste(wkdir,"Chrom.1",sep=""),
  col.names=c("position.morgans","chromosome"))
b1sim100.chrom2<-read.table(paste(wkdir,"Chrom.2",sep=""),
  col.names=c("position.morgans","chromosome"))
b1sim100.map<-list(chrom1=b1sim100.chrom1,chrom2=b1sim100.chrom2)
remove(list=c("b1sim100.chrom1","b1sim100.chrom2"))
b1sim100.qtl<-rqt1.import(paste(wkdir,"b1sim.qtl",sep=""))
b1sim100.qtl$d.MQ.haldane.Morgans<-d.haldane(b1sim100.qtl$recomb.MQ)

q.chrom1<-b1sim100.qtl[b1sim100.qtl$chromosome==1,]
q.chrom2<-b1sim100.qtl[b1sim100.qtl$chromosome==2,]
q.chrom1$position.morgans<-
  (b1sim100.map$chrom1[ q.chrom1$marker.M,"position.morgans"]
  +q.chrom1$d.MQ.haldane.Morgans)

```

```

q.chrom2$position.morgans<-
  (b1sim100.map$chrom2[ q.chrom2$marker.M,"position.morgans"]
   +q.chrom2$d.MQ.haldane.Morgans)
b1sim100.qtl<-list(chrom1=q.chrom1,chrom2=q.chrom2)

#-----
#   plot the genetic map on which the model is based
#-----

source("vuwfunc.r")
source("cro.import.r")
maxx<-max(c(b1sim100.map$chrom1$pos,b1sim100.map$chrom2$pos,
            b1sim100.qtl$chrom1$pos,b1sim100.map$chrom2$pos))
max1<-max(c(b1sim100.map$chrom1$pos,b1sim100.qtl$chrom1$pos))
max2<-max(c(b1sim100.map$chrom2$pos,b1sim100.map$chrom2$pos))
plot(c(3,0),c(0,maxx+0.1),axes=F, xlab="Chromosome",
     ylab="Position in Morgans",type="n")
title("Genetic Map")
axis(1,pos=-0.05, at=c(1,2))
axis(2,pos=0, at=seq(0,maxx,0.2))
lines( c(1,1),c(0,max1))
lines( c(2,2),c(0,max2))

chrom1.end<-length(b1sim100.map$chrom1[,1])
chrom2.end<-length(b1sim100.map$chrom2[,1])
labs1<-b1s1$markers[1:chrom1.end]
labs1q<-paste("QTL",b1sim100.qtl$chrom1[, "qtl.Q"])
text(b1sim100.map$chrom1$chromosome+0.05,b1sim100.map$chrom1$pos,labs1,adj=0)
#adj=0 means left justify, adj=1 means right justify
points(b1sim100.map$chrom1$chromosome,b1sim100.map$chrom1$pos,pch=3,cex=1.5)
text(b1sim100.qtl$chrom1$chromosome-0.05,b1sim100.qtl$chrom1$pos,labs1q,adj=1)
points(b1sim100.qtl$chrom1$chromosome,b1sim100.qtl$chrom1$pos,cex=0.8)

labs2<-b1s1$markers[(chrom1.end+1):(chrom1.end+chrom2.end)]
labs1q<-paste("QTL",b1sim100.qtl$chrom2[, "qtl.Q"])
text(b1sim100.map$chrom2$chromosome+0.05,b1sim100.map$chrom2$pos,labs2,adj=0)
points(b1sim100.map$chrom2$chromosome,b1sim100.map$chrom2$pos,pch=3,cex=1.5)
text(b1sim100.qtl$chrom2$chromosome-0.05,b1sim100.qtl$chrom2$pos,labs1q,adj=1)
points(b1sim100.qtl$chrom2$chromosome,b1sim100.qtl$chrom2$pos,cex=0.8)
remove(list=c("maxx","max1","max2","labs1","labs2"))

#b1s1$markers
n1<-length(b1sim100.map$chrom1[, "position.morgans"])
n2<-length(b1sim100.map$chrom2[, "position.morgans"])
d.curr.next1<-c(b1sim100.map$chrom1[2:n1,"position.morgans"]
               -b1sim100.map$chrom1[1:(n1-1),"position.morgans"],Inf)

```



```

d.curr.next2<-c(b1sim100.map$chrom2[2:n2,"position.morgans"]
               -b1sim100.map$chrom2[1:(n2-1),"position.morgans"],Inf)
r.curr.next<-c(r.haldane(d.curr.next1),r.haldane(d.curr.next2))
#save the marker map within each dataset
for (i in 1:nsamp){
  b1datafile<-paste("b1s",i,".cro1", sep="")
  b1object<-paste("b1s",i, sep="")
  assign(b1object,c(eval(as.name(b1object))),r.curr.next=list(r.curr.next)))
}
save(list=ls(),file="b1sim.RData")

#-----
#               Perform the analysis - part 1
# Search for QTL on chromosome 2, by Lander & Botstein interval mapping (IM),
# using the QTL cartographer module Zmapqtl: Model 3
#-----

#reformat the data for use with Zmapqtl
nsamp<-100
for (i in 1:nsamp){
  b1datafile<-paste("b1s",i,".cro1", sep="")
  b1datafile2<-paste("b1s",i,".cro", sep="")
  rcross.call<-paste("Rcross -A -V -W",wkdir,"-i", b1datafile,"-o",
                    b1datafile2, "-m b1sim.map -q b1sim.qtl -g 0")
  k<-system(rcross.call)
}

#Run Zmap QTL to search for QTL on chromosome 2 via interval mapping
set.seed(290) #want to use same seed for all data
myseed<-sample(1e6, 1)
for (i in 1:nsamp){
  b1datafile<-paste("b1s",i,".cro", sep="")
  b1Zfile<-paste("b1s",i,".IM.zed", sep="")
  b1Lfile<-paste("b1s",i,".el", sep="")
  b1Sfile<-paste("b1s",i,".es", sep="")
  zmapqtl.IM.call<-paste("Zmapqtl -A -V -W", wkdir, "-s", myseed,
                        "-e b1sim.IM.log -o", b1Zfile,"-i", b1datafile,"-m", "b1sim.map",
                        "-t 1","-l",b1Lfile, "-S",b1Sfile, "-M 3 -c 2 -d 1 -n 0 -w 10 -r 0 -b 0")
  k<-system(zmapqtl.IM.call)
}

#-----
#               Perform the analysis - part 2
# Zeng's Composite Interval Mapping (CIM) using QTL cartographer
# Fitting all the background markers: Model 1
#-----
nsamp<-100

```

```

wkdir<-"/u/students/nwill/newsims/"
set.seed(290) #want to use same seed for all data
myseed<-sample(1e6, 1)
for (i in 1:nsamp){
  b1datafile<-paste("b1s",i,".cro", sep="")
  b1Zfile<-paste("b1s",i,".CIM.zed", sep="")
  b1Lfile<-paste("b1s",i,".el", sep="")
  b1Sfile<-paste("b1s",i,".es", sep="")
  set.seed(290) #want to use same seed for all data
  myseed<-sample(1e6, 1)
  zmapqtl.CIM.call<-paste("Zmapqtl -A -V -W", wkdir, "-s", myseed,
    "-e b1sim.CIM.log -o" ,b1Zfile,"-i", b1datafile,"-m", "b1sim.map",
    "-t 1","-l",b1Lfile, "-S",b1Sfile,
    "-M 1 -c 2 -d 1 -n 0 -w 10 -r 0 -b 0")
  k<-system(zmapqtl.CIM.call) #R
}

#-----
# Import the results using the utility function zmapqtl.import(),
# then summarise ZmapQTL results for both IM and CIM .
#-----

wkdir<-"/u/students/nwill/newsims/"
nsamp<-100

zmap.all.intervals<-function(i,model="IM",chrom,marker.start,marker.end,wkdir,type="backcross"){
  b1Zfile<-paste("b1s",i,".",model,".zed", sep="")
  assign("zmap.out",zmapqtl.import(paste(wkdir,b1Zfile,sep=""),type),1)
  get.putative.qtl<-function(j,chrom){
    tmat<-zmap.out[(zmap.out[, "c"]==chrom)&(zmap.out[, "m"]==j),]
    biggest<-max(tmat[, "H0.H1"])
    qtl.test<-rbind(tmat[tmat[, "H0.H1"]==biggest,])
    qtl.test[1,]
  }
  val<-t(sapply(marker.start:marker.end,get.putative.qtl,chrom))
  val<-cbind(rep(i,length(val[,1])),val)
  dimnames(val)<-list(NULL,c("sample",dimnames(zmap.out)[[2]]))
  val
}

unlist.join.matrices<-function(zmap.data,nsamp,chrom,marker.start,marker.end){
  val<-zmap.data[[1]]
  for(i in 2:nsamp)
    val<-rbind(val,zmap.data[[i]])
  val
}

```

```

#At thei time, I am only searching chromosome 2 for QTL
chrom<-2
marker.start<-1
marker.end<-20

chrom2.zmap.IM <-lapply(1:nsamp,zmap.all.intervals,model="IM",
                        chrom,marker.start,marker.end,wkdir)
k<-unlist.join.matrices(chrom2.zmap.IM,nsamp,marker.start,marker.end)
IM.all<-lapply(marker.start:marker.end,function(h,k){k[k[, "m"]==h,]},k)
names(IM.all)<-paste("chrom2.",1:20,"to",2:21,sep="")

chrom2.zmap.CIM <-lapply(1:nsamp,zmap.all.intervals,model="CIM",
                        chrom,marker.start,marker.end,wkdir)
k<-unlist.join.matrices(chrom2.zmap.CIM,nsamp,marker.start,marker.end)
CIM.all<-lapply(marker.start:marker.end,function(h,k){k[k[, "m"]==h,]},k)
names(CIM.all)<-paste("chrom2.",1:20,"to",2:21,sep="")
remove(list=c("zmap.all.intervals", "unlist.join.matrices", "chrom2.zmap.IM",
              "chrom2.zmap.CIM", "k"))

zengtest<-function(h,nsamp,prob){
  #QTL Cartographer advocates using chi-square with one degree of freedom.
  #the user manual of QTL Cartographer Version 1.15 states:
  #'a value of H1/H0 of 3.84 or higher is evidence for a QTL'
  #returning percentage of times QTL detected in nsamp trials
  temp<-h[, "H0.H1"]
  critical.value<-qchisq(prob,1)
  length(temp[temp>critical.value])/nsamp*100
}

IM.detect.05<-sapply(IM.all,zengtest,nsamp,0.95)
CIM.detect.05<-sapply(CIM.all,zengtest,nsamp,0.95)
IM.detect.01<-sapply(IM.all,zengtest,nsamp,0.99)
CIM.detect.01<-sapply(CIM.all,zengtest,nsamp,0.99)
IM.detect.001<-sapply(IM.all,zengtest,nsamp,0.999)
CIM.detect.001<-sapply(CIM.all,zengtest,nsamp,0.999)

c2qtl.actual<-data.frame(QTL=rep(c("YES", "NO"),10),
                        recomb.MQ=rep(NA,20),additive.Q=rep(NA,20),dominance.Q=rep(NA,20))
c2qtl.actual[, "QTL"]<-"NO"
c2qtl.actual[q.chrom2$marker.M, "QTL"]<-"YES"
c2qtl.actual[q.chrom2$marker.M, "recomb.MQ"]<-q.chrom2$recomb.MQ
c2qtl.actual[q.chrom2$marker.M, c("additive.Q", "dominance.Q")]<-
  q.chrom2[, c("additive.Q", "dominance.Q")]
#dominance.Q= -2*dQQ (QTL Catographer manul page 37)
bQ.actual<- (c2qtl.actual[, "additive.Q"]- c2qtl.actual[, "dominance.Q"])

```

```

c2qtl.actual2<-data.frame(c2qtl.actual[,1:2],bQ.actual)
b1.IM.CIM.summary.05<-data.frame(IM.detect.05, CIM.detect.05,c2qtl.actual2)
b1.IM.CIM.summary.01<-data.frame(IM.detect.01, CIM.detect.01,c2qtl.actual2)
b1.IM.CIM.summary.001<-data.frame(IM.detect.001, CIM.detect.001,c2qtl.actual2)
save(list=c("CIM.all", "b1.IM.CIM.summary.05", "b1.IM.CIM.summary.01",
           "b1.IM.CIM.summary.001"), file="b1.IM.CIM.RData")

```

B.4 R code to implement the information matrix formulas for RIM1 and its sub-models

Table B.3: List of information matrix functions

Function	Description	Dependencies
emcov.fisher()	Calculates the Fisher information matrix in the context of the EM algorithm	model.config() mixing.probs() Dw.Dphi() D2w.Dphi2()
emcov.observed()	Calculates the conditional observed information matrix in the context of the EM algorithm	model.config() mixing.probs() Dw.Dphi()
model.config()	Identifies which model (RIM1=LQR, LQ, QR, CIM=Q, LR, L, R) is being fitted and hence the configuration of ϕ .	
mixing.probs()	Switching function for calculating the mixing proportions depending on the type of breeding design.	See Section B.1 for details.
Dw.Dphi()	Switching function for Dw.Dphi.b1() and Dw.Dphi.f2	Dw.Dphi.b1() Dw.Dphi.f2()
Dw.Dphi.b1()	Calculates the first partial derivative of $\ln \mathbf{W} = (\ln \mathbf{w})_{ik}$ with respect to ϕ for the backcross design.	index.genot()
Dw.Dphi.f2()	Calculates the first partial derivative of $\ln \mathbf{W} = (\ln \mathbf{w})_{ik}$ with respect to ϕ for the F2 design.	index.genot() df2g1() df2h1()

Table B.3: (continued)

Function	Description	Dependencies
df2h1()	Calculates the first partial derivative of $(\ln \mathbf{w})_{ik}$ with respect to ϕ for the F2 design, when Haldane's map function is assumed.	
df2g1()	Calculates the first partial derivative of $(\ln \mathbf{w})_{ik}$ with respect to ϕ for the F2 design, when a general three-locus map function is assumed.	
D2w.Dphi2()	Switching function for D2w.Dphi2.b1() and D2w.Dphi2.f2()	D2w.Dphi2.b1() D2w.Dphi2.f2()
D2w.Dphi2.b1()	Calculates the second partial derivative of $\ln \mathbf{W} = (\ln \mathbf{w})_{ik}$ with respect to ϕ for the backcross design.	index.genot()
D2w.Dphi2.f2()	Calculates the second partial derivative of $\ln \mathbf{W} = (\ln \mathbf{w})_{ik}$ with respect to ϕ for the F2 design.	index.genot() d2f2g1() d2f2h1()
d2f2h1()	Calculates the second partial derivative of $(\ln \mathbf{w})_{ik}$ with respect to ϕ for the F2 design, when Haldane's map function is assumed.	
d2f2g1()	Calculates the second partial derivative of $(\ln \mathbf{w})_{ik}$ with respect to ϕ for the F2 design, when a general three-locus map function is assumed.	
index.genot()	Identify marker genotypes in the model.	See Section B.1 for details.

Source Code

```
#-----
# emcov.fisher()
#-----
emcov.fisher<-function(chosen.model,cross,hypothesis,mapfun,genot,genot2,markerg,params,X2,Ce,
                      cofactors.names,n,nqgen,nmgen,recomb,cov=FALSE){
  N<-sum(n)
  probs<-params$probs #the mixing proportions
  probs2<-probs
  if (mapfun=="Haldane"){
    probs2["pQ1"]<-recomb["pQ1"]
  }
}
```

```

    probs2["pL1"]<-recomb["pL1"]
    probs2["pR1"]<-recomb["pR1"]
  }
  m<-model.config(cross,mapfun,chosen.model,probs2,markerg,genot,genot2,recomb)
  w<-mixing.probs(cross,hypothesis,chosen.model,probs2,markerg,genot,genot2,nqgen)
  d2lnw<-D2w.Dphi2(cross,mapfun,chosen.model,probs,markerg,genot,genot2,nqgen,m)
  sigma2<-params$variance
  main.names<-dimnames(Ce)[[2]]
  B<-params$effects[main.names,]
  Bstar<-params$effects[cofactors.names,]
  length.phi<-length(probs)
  length.B<-length(c(B))
  length.Bstar<-length(c(Bstar))

  #We need an index to identify the marker groups
  ind<-vector("list", nmgen)
  names(ind)<-names(n)
  ind[[1]]<-1:n[1]
  for(i in 2:nmgen)
    ind[[i]]<- (1+sum(n[1:(i-1)])):sum(n[1:i])
  Z<-matrix(0,nrow=N,ncol=nqgen)
  I.sigma2.sigma2<-N/(2*sigma2^2)
  I.phi.phi<-matrix(0,nrow=length.phi,ncol=length.phi)
  mconfig<-genot2$mconfig

  for(i in 1:nmgen){
    Z[ind[[i]], ]<-cbind(rep(1,n[i])) %*% w[i,]
    Zi<-rbind(Z[ind[[i]], ])
    iQ<-NULL
    iR<-NULL
    iL<-NULL
    if (length(m$Qfit)>0){
      if (mapfun=="Haldane")
        iQ<-d2lnw[[i]]$dpQ2 %*% t(Zi) %*% rep(1,n[i])
      else
        iQ<-cbind(d2lnw[[i]]$dpQ1 %*% t(Zi) %*% rep(1,n[i]),
                  d2lnw[[i]]$dpQ2 %*% t(Zi) %*% rep(1,n[i]) )
    }
    if (length(m$Lfit)>0){
      if (mconfig["K"]==1){
        if (mapfun=="Haldane")
          iL<-d2lnw[[i]]$dpL2 %*% t(Zi) %*% rep(1,n[i])
        else
          iL<-cbind(d2lnw[[i]]$dpL1 %*% t(Zi) %*% rep(1,n[i]),
                    d2lnw[[i]]$dpL2 %*% t(Zi) %*% rep(1,n[i]) )
      }
    }
  }

```

```

    }
    else iL<-d2lnw[[i]]$dpL *** t(Zi)*** rep(1,n[i])
  }
  if (length(m$Rfit)>0){
    if (mconfig["0"]==1){
      if (mapfun=="Haldane")
        iR<-d2lnw[[i]]$dpR2 *** t(Zi) *** rep(1,n[i])
      else
        iR<-cbind(d2lnw[[i]]$dpR1 *** t(Zi) *** rep(1,n[i]),
                  d2lnw[[i]]$dpR2 *** t(Zi) *** rep(1,n[i]) )
    }
    else
      iR<-d2lnw[[i]]$dpR *** t(Zi) *** rep(1,n[i])
  }
  I.phi.phi<- I.phi.phi - cbind(iL,iQ,iR)
}

I.beta.beta<-1/sigma2 * t(Ce)***diag(c(rep(1,N)***Z)***Ce
I.beta.betastar<-1/sigma2 * t(Ce)***t(Z)***X2
I.betastar.betastar<-1/sigma2 * t(X2)***X2
I.beta.phi<-matrix(0,nrow=length.B,ncol=length.phi)
I.beta.sigma2<-matrix(0,nrow=length.B,ncol=1)
I.betastar.phi<-matrix(0,nrow=length.Bstar,ncol=length.phi)
I.betastar.sigma2<-matrix(0,nrow=length.Bstar,ncol=1)
I.sigma2.phi<-matrix(0, nrow=1,ncol=length.phi)

imat<-rbind( cbind(I.beta.beta,I.beta.betastar,I.beta.sigma2,I.beta.phi),
             cbind(t(I.beta.betastar),I.betastar.betastar,I.betastar.sigma2,I.betastar.phi),
             cbind(t(I.beta.sigma2), t(I.betastar.sigma2),I.sigma2.sigma2,I.sigma2.phi),
             cbind(t(I.beta.phi), t(I.betastar.phi),t(I.sigma2.phi), I.phi.phi))

remove(list=c("I.beta.beta","I.beta.betastar","I.beta.sigma2", "I.beta.phi","I.betastar.betastar",
              "I.betastar.sigma2","I.betastar.phi","I.sigma2.sigma2","I.sigma2.phi", "I.phi.phi"))

infmat.singular<-FALSE
imat.inv<- try(solve(imat),silent=TRUE)
if (inherits(imat.inv, "try-error")){
  infmat.singular<-TRUE
  gotMASS<- try(find(ginv, mode="function"),silent=TRUE)
  if (inherits(gotMASS, "try-error"))
    require("MASS")
  imat.inv<-ginv(imat)
}

param.names<-c(dimnames(params$effects)[[1]],"sigma2",names(params$probs))

```

```

dimnames(imat.inv)<-list(param.names,param.names)
var.all<-diag(imat.inv)
std.err.beta<-sqrt(var.all[dimnames(params$effects)[[1]]])
zstat0.beta<-params$effects[,1]/std.err.beta
pval.beta<-2*(1-pnorm(abs(zstat0.beta)))

std.err.probs<-sqrt(var.all[names(params$probs)])
zstat0.probs<-params$probs/std.err.probs
zstat1.probs<-(params$probs-1)/std.err.probs
pval0.probs<-1-pnorm(zstat0.probs)
pval1.probs<-pnorm(zstat1.probs)
b.result<-cbind(std.err.beta,zstat0.beta,pval.beta)
dimnames(b.result)<-list(dimnames(params$effects)[[1]],c("std.err","z0","P>|z0|"))
probs.result<-cbind(std.err.probs,zstat0.probs,zstat1.probs,pval0.probs,pval1.probs )
dimnames(probs.result)<-list(names(params$probs),c("std.err","z0","z1","P>z0","P<z1"))
val<- list(b.result=b.result, probs.result=probs.result,
           infmat.singular= infmat.singular)
if (cov==TRUE)
  val$cov<-imat.inv

val
}

#-----
# emcov.observed()
#-----
emcov.observed<-function(chosen.model,cross,hypothesis,mapfun,genot, genot2,markerg,params,Z,X2,Ce,
                          cofactors.names, n,nqgen,nmgen,y,recomb,cov=FALSE){
  N<-sum(n)
  probs<-params$probs #the mixing proportions
  probs2<-probs
  if (mapfun=="Haldane"){
    probs2["pQ1"]<-recomb["pQ1"]
    probs2["pL1"]<-recomb["pL1"]
    probs2["pR1"]<-recomb["pR1"]
  }
  m<-model.config(cross,mapfun,chosen.model,probs2,markerg,genot,genot2,recomb)
  w<-mixing.probs(cross,hypothesis,chosen.model,probs2,markerg,genot,genot2,nqgen)
  sigma2<-params$variance
  #We need an index to identify the marker groups
  ind<-vector("list", nmgen)
  names(ind)<-names(n)
  ind[[1]]<-1:n[1]
  for(i in 2:nmgen)
    ind[[i]]<- (1+sum(n[1:(i-1)])):sum(n[1:i])
}

```



```

main.names<-dimnames(Ce)[[2]]
B<-params$effects[main.names,]
Bstar<-params$effects[cofactors.names,]
length.phi<-length(probs)
length.B<-length(c(B))
length.Bstar<-length(c(Bstar))

mu<-Ce%*%B #mu.qtl
mu.cofactors<-as.list(n)
for(i in 1:nngen)
  mu.cofactors[[i]]<-X2[ind[[i]],]%*%Bstar
mu.cofactors<-unlist(mu.cofactors)
#Y will be a N vector of trait values partitioned by marker group
Y<-unlist(y) #uppercase Y is a numeric vector, lowercase y is a list
Y<- (Y - mu.cofactors) #center y relative to the cofactors

#Now calculate the components of the observed information matrix
diag2.1n.Z<-diag(c(rep(1,N)%*%Z))
diag3.yt.diag1yt.Z<-diag(c(t(Y)%*%diag(Y)%*%Z))
diag1.mu<-diag(c(mu))
diag1.yt<-diag(c(Y))
diag2.yt.Z<-diag(c(t(Y)%*%Z))

I.beta.beta<-(1/sigma2 * t(Ce)%*%diag2.1n.Z%*%Ce
-1/(sigma2^2) * t(Ce)%*%(diag3.yt.diag1yt.Z
+ diag1.mu %*% (diag2.1n.Z %*% diag1.mu - 2*diag2.yt.Z)
- (diag1.mu %*% t(Z) -t(Z)%*% diag1.yt)%*%
t((diag1.mu %*% t(Z) -t(Z)%*% diag1.yt)))%*%Ce
)

I.sigma2.sigma2<- ( 1/(sigma2^3)*(t(Y)%*%Y - 2*t(Y)%*%Z%*%mu
+ t(mu)%*%diag2.1n.Z %*% mu )
- N/(2*sigma2^2)
- 1/(4*sigma2^4) %*% t(mu) %*% ( 4*diag3.yt.diag1yt.Z
- 4* t(Z) %*% diag1.yt %*% diag1.yt %*% Z
+ 4* diag1.mu %*% (diag2.yt.Z - t(Z)%*% diag1.yt %*% Z)
+ diag1.mu %*% (diag2.1n.Z -t(Z) %*% Z) %*% diag1.mu
) %*% mu
)

mconfig<-genot2$mconfig
d2lnw<-D2w.Dphi2(cross,mapfun,chosen.model,probs,markerg,genot,genot2,nngen,m)
I.phi.phi.Ic<-matrix(0,nrow=length.phi,ncol=length.phi)
for(i in 1:nngen){
  iQ<-NULL

```

```

iR<-NULL
iL<-NULL
Zi<-rbind(Z[ind[[i]], ])
if (length(m$Qfit)>0){
  if (mapfun=="Haldane")
    iQ<-d2lnw[[i]]$dpQ2 %*% t(Zi) %*% rep(1,n[i])
  else
    iQ<-cbind(d2lnw[[i]]$dpQ1 %*% t(Zi) %*% rep(1,n[i]),
              d2lnw[[i]]$dpQ2 %*% t(Zi) %*% rep(1,n[i]) )
}
if (length(m$Lfit)>0){
  if (mconfig["K"]==1){
    if (mapfun=="Haldane")
      iL<-d2lnw[[i]]$dpL2 %*% t(Zi) %*% rep(1,n[i])
    else
      iL<-cbind(d2lnw[[i]]$dpL1 %*% t(Zi) %*% rep(1,n[i]),
                d2lnw[[i]]$dpL2 %*% t(Zi) %*% rep(1,n[i]) )
  }
  else iL<-d2lnw[[i]]$dpL %*% t(Zi)%*% rep(1,n[i])
}
if (length(m$Rfit)>0){
  if (mconfig["O"]==1){
    if (mapfun=="Haldane")
      iR<-d2lnw[[i]]$dpR2 %*% t(Zi) %*% rep(1,n[i])
    else
      iR<-cbind(d2lnw[[i]]$dpR1 %*% t(Zi) %*% rep(1,n[i]),
                d2lnw[[i]]$dpR2 %*% t(Zi) %*% rep(1,n[i]) )
  }
  else
    iR<-d2lnw[[i]]$dpR %*% t(Zi) %*% rep(1,n[i])
}
I.phi.phi.Ic<- I.phi.phi.Ic - cbind(iL,iQ,iR)
}
d1lnw<-Dw.Dphi(cross,mapfun,chosen.model,probs,markerg,genot,genot2,nqgen,m)
I.phi.phi.Im<-matrix(0,nrow=length.phi,ncol=length.phi)
for(i in 1:nngen)
  for(j in 1:nngen){
    Di<- rbind(diag(rep(1,N))[ind[[i]],])
    Dj<- rbind(diag(rep(1,N))[ind[[j]],])
    Zi<-rbind(Z[ind[[i]], ])
    Zj<-rbind(Z[ind[[j]], ])
    I.phi.phi.Im <- (I.phi.phi.Im +
                     t(d1lnw[[i]])%*% (diag (c(rep(1,n[j]))%*% Dj %*% t(Di) %*% Zi))
                     - t(Zi) %*% Di %*% t(Dj)%*% Zj )%*% d1lnw[[j]] )
  }
}

```

```

I.phi.phi <- ( I.phi.phi.Ic - I.phi.phi.Im)
remove(list=c("I.phi.phi.Ic", "I.phi.phi.Im", "Di", "Dj"))

I.beta.sigma2<-(-1/sigma2^2 * t(Ce)%*(diag2.1n.Z%*mu -t(Z)%*Y)
-1/(2*sigma2^3) * t(Ce)%*
( 2*diag1.mu %*( diag2.yt.Z - t(Z)%*diag1.yt%*Z)
-2*diag3.yt.diag1yt.Z + 2*t(Z)%*diag1.yt%*diag1.yt%*Z
-diag1.mu %*(diag2.1n.Z - t(Z)%*Z) %* diag1.mu
+ (diag2.yt.Z - t(Z)%*diag1.yt%*Z)%*diag1.mu)%*mu )

I.beta.phi<-matrix(0,nrow=length.B,ncol=length.phi)
I.sigma2.phi<-matrix(0, nrow=1,ncol=length.phi)
for(i in 1:nmgen){
  Di<-rbind(diag(rep(1,N))[ind[[i]],])
  Zi<-rbind(Z[ind[[i]],])
  diag3.1ni.Zi.diag1mu<-diag(c(rep(1,n[i])%*Zi%*diag1.mu) )
  diag3.yt.Dit.Zi<-diag(c(t(Y)%*t(Di)%*Zi) )
  Zt.diag1y.Dit.Zi<-t(Z) %* diag1.yt %* t(Di)%*Zi
  I.beta.phi <- (I.beta.phi +
    1/sigma2 * t(Ce)%*((diag3.1ni.Zi.diag1mu - diag1.mu %*
      t(Zi) %* Zi - diag3.yt.Dit.Zi + Zt.diag1y.Dit.Zi)%* d1lnw[[i]]) )

  I.sigma2.phi <- (I.sigma2.phi +
    (-1)/(2*sigma2^2) * t(mu)%*((diag3.1ni.Zi.diag1mu - diag1.mu %*
      t(Zi) %* Zi - 2* diag3.yt.Dit.Zi + 2*Zt.diag1y.Dit.Zi)%* d1lnw[[i]]) )
}
remove(list=c("Di", "Zi", "diag3.1ni.Zi.diag1mu", "diag3.yt.Dit.Zi", "Zt.diag1y.Dit.Zi"))

diag2.mut.Zt<-diag(c(t(mu)%*t(Z)))
I.beta.betastar<- (1/sigma2 * t(Ce)%*t(Z)%*X2
-1/(sigma2^2) * t(Ce)%*(diag1.mu %*(diag1.mu %* t(Z) - t(Z)%*diag1.yt)
-(diag1.mu %* t(Z) - t(Z)%*diag1.yt)%*diag2.mut.Zt )%*X2 )

diag3.mut.diag1mu.Zt <-diag(c(t(mu) %* diag1.mu %* t(Z)))
I.betastar.betastar<- ( 1/sigma2 * t(X2)%*X2 -1/(sigma2^2) * t(X2)%*(diag3.mut.diag1mu.Zt
- diag2.mut.Zt %* diag2.mut.Zt)%*X2 )

I.betastar.sigma2<- (1/(sigma2^2)*t(X2)%*(Z%*mu-Y)
-1/(2*sigma2^3)*t(X2)%*( 2*diag3.mut.diag1mu.Zt %* Y
-2*diag2.mut.Zt %* diag2.mut.Zt %* Y
-Z %* diag1.mu %* diag1.mu %* mu
+ diag2.mut.Zt %* Z %* diag1.mu %* mu) )

I.betastar.phi<-matrix(0,nrow=length.Bstar,ncol=length.phi)
for(i in 1:nmgen){

```

```

Di<-rbind(diag(rep(1,N))[ind[[i]],])
diag2.1ni.Di<-rbind(diag(c(rep(1,n[i]))%*%Di) ))
I.betastar.phi <- (I.betastar.phi +
  1/sigma2 * t(X2)%*%(( diag2.mut.Zt %*% diag2.1ni.Di %*% Z
    -diag2.1ni.Di %*% Z %*% diag1.mu )%*% d1lnw[[i]] )
)
remove(list=c("Di","diag2.1ni.Di","diag2.1n.Z","diag3.yt.diag1yt.Z",
  "diag1.mu","diag1.yt", "diag2.yt.Z","diag2.mut.Zt" ))

imat<-rbind( cbind(I.beta.beta,I.beta.betastar,I.beta.sigma2,I.beta.phi),
  cbind(t(I.beta.betastar),I.betastar.betastar,I.betastar.sigma2,I.betastar.phi),
  cbind(t(I.beta.sigma2), t(I.betastar.sigma2),I.sigma2.sigma2,I.sigma2.phi),
  cbind(t(I.beta.phi), t(I.betastar.phi),t(I.sigma2.phi), I.phi.phi) )
remove(list=c("I.beta.beta","I.beta.betastar","I.beta.sigma2", "I.beta.phi","I.betastar.betastar",
  "I.betastar.sigma2","I.betastar.phi","I.sigma2.sigma2","I.sigma2.phi", "I.phi.phi"))

# If imat is not positive definite then find a positive definite submatrix of imat.
# The covariance matrix will be constructed by taking the inverse this positive definite submatrix
# and setting the remaining rows and columns of imat.inv to zero.
imat.len<-length(imat[1,])
eigen.imat<-try(eigen(imat,TRUE,TRUE),silent=TRUE)
if (inherits(eigen.imat, "try-error"))
  imat.is.negative.definite<-TRUE
else #check eigenvalues
  imat.is.negative.definite<-any(eigen.imat$values<0)

i<-imat.len
minlen<-length(params$effects[,1])
while((imat.is.negative.definite) & (i>=minlen)){
  i<-i-1
  imat<-imat[1:i,1:i]
  eigen.values.imat<-try(eigen(imat,TRUE,TRUE),silent=TRUE)
  if (inherits(eigen.imat, "try-error"))
    imat.is.negative.definite<-TRUE
  else
    imat.is.negative.definite<-any(eigen.imat$values<0)
}
imat2.len<-length(imat[1,])

infmat.singular<-FALSE
imat2.inv<- try(solve(imat),silent=TRUE)
if (inherits(imat2.inv, "try-error")){
  infmat.singular<-TRUE
  gotMASS<- try(find(ginv, mode="function"),silent=TRUE)
  if (inherits(gotMASS, "try-error"))

```

```

        require("MASS")
        imat2.inv<-ginv(imat)
    }
    imat.inv<-matrix(NA,nrow=imat.len,ncol=imat.len)
    imat.inv[1:imat2.len,1:imat2.len]<-imat2.inv
    param.names<-c(dimnames(params$effects)[[1]],"sigma2",names(params$probs))
    if (imat2.len==imat.len)
        dropped<-"None"
    else
        dropped<-param.names[(imat2.len+1):imat.len]
    dimnames(imat.inv)<-list(param.names,param.names)

    var.all<-diag(imat.inv)
    std.err.beta<-sqrt(var.all[dimnames(params$effects)[[1]]])
    zstat0.beta<-params$effects[,1]/std.err.beta
    pval.beta<-2*(1-pnorm(abs(zstat0.beta)))
    std.err.probs<-sqrt(var.all[names(params$probs)])
    zstat0.probs<-params$probs/std.err.probs
    zstat1.probs<-(params$probs-1)/std.err.probs
    pval0.probs<-1-pnorm(zstat0.probs)
    pval1.probs<-pnorm(zstat1.probs)

    b.result<-cbind(std.err.beta,zstat0.beta,pval.beta)
    dimnames(b.result)<-list(dimnames(params$effects)[[1]], c("std.err","z0","P>|z0|"))
    probs.result<-cbind(std.err.probs,zstat0.probs,zstat1.probs,pval0.probs,pval1.probs )
    dimnames(probs.result)<-list(names(params$probs), c("std.err","z0","z1","P>z0","P<z1"))
    val<- list(b.result=b.result, probs.result=probs.result, infmat.singular= infmat.singular)
    if (cov==TRUE)
        val$cov<-imat.inv
    val
}

#-----
# model.config()
#-----
model.config<-function(cross,mapfun,chosen.model,probs,markerg,genot,genot2,rmn){
    #this function determines which of the 7 models is being fitted.
    Qfit<-grep("Q",chosen.model)
    Rfit<-grep("R",chosen.model)
    Lfit<-grep("L",chosen.model)

    xgen<-NULL
    qt13<-genot2$qt13
    highL<-grep("LL",qt13)
    highR<-grep("RR",qt13)

```

```

highQ<-grep("QQ",qt13)
if (cross=="F2"){
  hetL<-grep("Ll",qt13)
  hetR<-grep("Rr",qt13)
  hetQ<-grep("Qq",qt13)
}
mconfig<-genot2$mconfig

pQ1<-pQ2<-0
if(length(Qfit)==0){
  if (cross=="F2")
    xgen<-c(highQ,hetQ)
  else
    xgen<-highQ
}
else{
  pQ2<-probs["pQ2"]
  pQ1<-probs["pQ1"]
}

pL1<-pL2<-pL<-0
if(length(Lfit)==0){
  if (cross=="F2")
    xgen<-c(highL,hetL,xgen)
  else
    xgen<-c(highL,xgen)
}
else if (mconfig["K"]==1){
  pL1<-probs["pL1"]
  pL2<-probs["pL2"]
}
else pL<-probs["pL"]

pR1<-pR2<-pR<-0
if(length(Rfit)==0){
  if (cross=="F2")
    xgen<-c(highR,hetR,xgen)
  else
    xgen<-c(highR,xgen)
}
else if (mconfig["0"]==1){
  pR1<-probs["pR1"]
  pR2<-probs["pR2"]
}

```

```

else pR<-probs["pR"]

kgen<-(1:length(qt13))

if (length(xgen)>0){
  xgen<-unique(xgen)
  kgen<-kgen[-xgen]
}

list(Lfit=Lfit,Rfit=Rfit,Qfit=Qfit,pQ1=pQ1,pQ2=pQ2,
     pL1=pL1,pL2=pL2,pL=pL,pR1=pR1,pR2=pR2,pR=pR,kgen=kgen)
}

#-----
# Dw.Dphi.b1()
#-----
Dw.Dphi.b1<-function(mapfun,chosen.model,probs,markerg,genot,genot2,nqgen,m){
  pQ1<-m$pQ1
  pQ2<-m$pQ2
  pL1<-m$pL1
  pL2<-m$pL2
  pR1<-m$pR1
  pR2<-m$pR2
  pL<-m$pL
  pR<-m$pR
  mconfig<-genot2$mconfig
  rkm<-genot2$rkm
  rmn<-genot2$rmn
  rno<-genot2$rno

  dfit.names<-c("dPL","dPR","dPL1","dPL2","dPR1","dPR2","dpQ1","dpQ2")
  names(dfit.names)<-c("pL","pR","pL1","pL2","pR1","pR2","pQ1","pQ2")
  dfit.names<-dfit.names[names(probs)]
  mat<-matrix(0,nrow=nqgen,ncol=length(dfit.names))
  dimnames(mat)<-list(NULL,dfit.names)
  d1lnw<-lapply(1:length(genot),function(h,x){x},mat)
  names(d1lnw)<-genot

  matlist<-function(mylist,dqt1,index.qtl,val){
    lapply(mylist,function(x,dqt1,index.qtl,val){
      x[,dqt1]<-val[index.qtl]; x},
      dqt1,index.qtl,val)
  }
}

```

```

if (length(m$Lfit)>0) {
  if (mconfig["K"]==1){
    gg<-index.genot(cross="B1",M="K",N="M",genot2=genot2)
  if (mapfun=="Haldane"){
    dpL1.dpL2 <- (1-2*pL2)*rkm^2 /((1-2*pL1)*(1-rkm)^2)
    d1lnw[gg$KKMM]<-matlist(d1lnw[gg$KKMM], "dpL2",m$kgen,(rep(c(1/pL1,-1/(1-pL1)),c(4,4))*dpL1.dpL2))
    d1lnw[gg$KKMm]<-matlist(d1lnw[gg$KKMm], "dpL2",m$kgen,rep(c(1/pL2,-1/(1-pL2)),c(4,4)))
    d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm], "dpL2",m$kgen,
      rev(rep(c(1/pL1,-1/(1-pL1)),c(4,4))*dpL1.dpL2))
    d1lnw[gg$KkMM]<- matlist(d1lnw[gg$KkMM], "dpL2",m$kgen,rev(rep(c(1/pL2,-1/(1-pL2)), c(4,4))))
  }
  else{
    d1lnw[gg$KKMM]<- matlist(d1lnw[gg$KKMM], "dpL1",m$kgen,rep(c(1/pL1,-1/(1-pL1)), c(4,4)))
    d1lnw[gg$KKMm]<- matlist(d1lnw[gg$KKMm], "dpL2",m$kgen,rep(c(1/pL2,-1/(1-pL2)), c(4,4)))
    d1lnw[gg$KkMm]<- matlist(d1lnw[gg$KkMm], "dpL1",m$kgen,rev(rep(c(1/pL1,-1/(1-pL1)), c(4,4))))
    d1lnw[gg$KkMM]<- matlist(d1lnw[gg$KkMM], "dpL2",m$kgen,rev(rep(c(1/pL2,-1/(1-pL2)), c(4,4))))
  }
}
else{
  gg<-index.genot(cross="B1",M="M",genot2=genot2)
  d1lnw[gg$MM]<- matlist(d1lnw[gg$MM], "dpL",m$kgen,rep(c(1/pL,-1/(1-pL)), c(4,4)))
  d1lnw[gg$Mm]<- matlist(d1lnw[gg$Mm], "dpL",m$kgen,rev(rep(c(1/pL,-1/(1-pL)), c(4,4))))
}
}

if (length(m$Rfit)>0) {
  if (mconfig["O"]==1){
    gg<-index.genot(cross="B1",M="N",N="O",genot2=genot2)
    if (mapfun=="Haldane"){
      dpR1.dpR2 <- (1-2*pR2)*rno^2 /((1-2*pR1)*(1-rno)^2)
      d1lnw[gg$NNOO]<- matlist(d1lnw[gg$NNOO], "dpR2",m$kgen,(rep(c(1/pR1,-1/(1-pR1)),4)*dpR1.dpR2))
      d1lnw[gg$NNOo]<- matlist(d1lnw[gg$NNOo], "dpR2",m$kgen,rep(c(1/pR2,-1/(1-pR2)),4))
      d1lnw[gg$NnOo]<- matlist(d1lnw[gg$NnOo], "dpR2",m$kgen,rev(rep(c(1/pR1,-1/(1-pR1)),4)*dpR1.dpR2))
      d1lnw[gg$NnOO]<- matlist(d1lnw[gg$NnOO], "dpR2",m$kgen,rev(rep(c(1/pR2,-1/(1-pR2)),4)))
    }
    else{
      d1lnw[gg$NNOO]<- matlist(d1lnw[gg$NNOO], "dpR1",m$kgen,rep(c(1/pR1,-1/(1-pR1)),4))
      d1lnw[gg$NNOo]<- matlist(d1lnw[gg$NNOo], "dpR2",m$kgen,rep(c(1/pR2,-1/(1-pR2)),4))
      d1lnw[gg$NnOO]<- matlist(d1lnw[gg$NnOO], "dpR2",m$kgen,rev(rep(c(1/pR2,-1/(1-pR2)),4)))
      d1lnw[gg$NnOo]<- matlist(d1lnw[gg$NnOo], "dpR1",m$kgen,rev(rep(c(1/pR1,-1/(1-pR1)),4)))
    }
  }
  else{
    gg<-index.genot(cross="B1",N="N",genot2=genot2)
    d1lnw[gg$NN]<- matlist(d1lnw[gg$NN], "dpR",m$kgen,rep(c(1/pR,-1/(1-pR)),4))
  }
}

```



```

      d1lnw[gg$Nn]<- matlist(d1lnw[gg$Nn], "dpR", m$kgen, rev(rep(c(1/pR, -1/(1-pR)), 4)))
    }
  }
  if(length(m$Qfit)>0) {
    gg<-index.genot(cross="F2", M="M", N="N", genot2=genot2)
    if (mapfun=="Haldane"){
      dpQ1.dpQ2 <- (1-2*pQ2)*rmn^2 / ((1-2*pQ1)*(1-rmn)^2)
      d1lnw[gg$MMNN]<-matlist(d1lnw[gg$MMNN], "dpQ2", m$kgen,
        (rep(c(1/pQ1, 1/pQ1, -1/(1-pQ1), -1/(1-pQ1)), 2 ) * dpQ1.dpQ2))
      d1lnw[gg$MMNn]<-matlist(d1lnw[gg$MMNn], "dpQ2", m$kgen, rep(c(1/pQ2, 1/pQ2, -1/(1-pQ2), -1/(1-pQ2)), 2))
      d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ2", m$kgen,
        rev(rep(c(1/pQ1, 1/pQ1, -1/(1-pQ1), -1/(1-pQ1)), 2 ) * dpQ1.dpQ2))
      d1lnw[gg$MmNN]<-matlist(d1lnw[gg$MmNN], "dpQ2", m$kgen,
        rev(rep(c(1/pQ2, 1/pQ2, -1/(1-pQ2), -1/(1-pQ2)), 2)))
    }
    else{
      d1lnw[gg$MMNN]<-matlist(d1lnw[gg$MMNN], "dpQ1", m$kgen, rep(c(1/pQ1, 1/pQ1, -1/(1-pQ1), -1/(1-pQ1)), 2))
      d1lnw[gg$MMNn]<-matlist(d1lnw[gg$MMNn], "dpQ2", m$kgen, rep(c(1/pQ2, 1/pQ2, -1/(1-pQ2), -1/(1-pQ2)), 2))
      d1lnw[gg$MmNn]<- matlist(d1lnw[gg$MmNn], "dpQ1", m$kgen,
        rev(rep(c(1/pQ1, 1/pQ1, -1/(1-pQ1), -1/(1-pQ1)), 2)))
      d1lnw[gg$MmNN]<- matlist(d1lnw[gg$MmNN], "dpQ2", m$kgen,
        rev(rep(c(1/pQ2, 1/pQ2, -1/(1-pQ2), -1/(1-pQ2)), 2 )))
    }
  }
}

d1lnw[markerg]
}

#-----
# D2w.Dphi2.b1()
#-----
D2w.Dphi2.b1<-function(mapfun, chosen.model, probs, markerg, genot, genot2, nqgen, m){
  pQ1<-m$pQ1
  pQ2<-m$pQ2
  pL1<-m$pL1
  pL2<-m$pL2
  pR1<-m$pR1
  pR2<-m$pR2
  pL<-m$pL
  pR<-m$pR

  mconfig<-genot2$mconfig
  rkm<-genot2$rkm
  rmn<-genot2$rmn
  rno<-genot2$rno

```

```

dfit.names<-c("dpL","dpR","dpL1","dpL2","dpR1","dpR2","dpQ1","dpQ2")
names(dfit.names)<-c("pL","pR","pL1","pL2","pR1","pR2","pQ1","pQ2")
dfit.names<-dfit.names[names(probs)]
mat<-matrix(0,nrow=length(dfit.names),ncol=nqgen)
dimnames(mat)<-list(dfit.names,NULL)
d2h1<-as.list(NULL)
d2h1<-lapply(1:length(dfit.names),function(h,x){x},mat)
names(d2h1)<-dfit.names
d2lnw<-lapply(1:length(genot),function(h,x){x},d2h1)
names(d2lnw)<-genot

#indexing a three-level nested list
#using the fact that dpR dpL=0 etc
matlist<-function(mylist,dqtl,index.qtl,val){
  lapply(mylist,function(x,dqtl,index.qtl,val){
    x[[dqtl]][dqtl,<-val[index.qtl]; x},
    dqtl,index.qtl,val)
  }
}

if(length(m$Lfit)>0) {
  if (mconfig["K"]==1){
    gg<-index.genot(cross="B1",M="K",N="M",genot2=genot2)
    d2h1.dpL12<- (-1)*rep(c(1/pL1^2,1/(1-pL1)^2),c(4,4))
    d2h1.dpL22<- (-1)*rep(c(1/pL2^2,1/(1-pL2)^2),c(4,4))

    if (mapfun=="Haldane"){
      dpL1.dpL2 <- (1-2*pL2)*rkm^2 /((1-2*pL1)*(1-rkm)^2)
      d2pL1.dpL22<- (-2)*rkm^2 /((1-2*pL1)*(1-rkm)^2)
      dh1.dpL1<-rep(c(1/pL1,-1/(1-pL1)), c(4,4))

      tempi<-(d2h1.dpL12 * (dpL1.dpL2)^2 + dh1.dpL1 * d2pL1.dpL22)

      d2lnw[gg$KKMM]<- matlist(d2lnw[gg$KKMM],"dpL2",m$kgen,tempi)
      d2lnw[gg$KKMm]<- matlist(d2lnw[gg$KKMm],"dpL2",m$kgen,d2h1.dpL22)
      d2lnw[gg$KkMm]<- matlist(d2lnw[gg$KkMm],"dpL2",m$kgen,rev(tempi))
      d2lnw[gg$KkMM]<- matlist(d2lnw[gg$KkMM],"dpL2",m$kgen,rev(d2h1.dpL22))
    }
  }
  else{
    d2lnw[gg$KKMM]<- matlist(d2lnw[gg$KKMM],"dpL1",m$kgen,d2h1.dpL12)
    d2lnw[gg$KKMm]<- matlist(d2lnw[gg$KKMm],"dpL2",m$kgen,d2h1.dpL22)
    d2lnw[gg$KkMM]<- matlist(d2lnw[gg$KkMM],"dpL2",m$kgen,rev(d2h1.dpL22))
    d2lnw[gg$KkMm]<- matlist(d2lnw[gg$KkMm],"dpL1",m$kgen,rev(d2h1.dpL12))
  }
}
}

```

```

else{
  gg<-index.genot(cross="B1",M="M",genot2=genot2)
  d2lnw[gg$MM]<- matlist(d2lnw[gg$MM], "dpL",m$kgen,(-1)*rep(c(1/pL^2,1/(1-pL)^2),c(4,4)))
  d2lnw[gg$Mm]<- matlist(d2lnw[gg$Mm], "dpL",m$kgen,rev((-1)*rep(c(1/pL^2,1/(1-pL)^2),c(4,4))))
}
}
if(length(m$Rfit)>0) {
  if (mconfig["0"]==1){
    gg<-index.genot(cross="B1",M="N",N="0",genot2=genot2)
    d2h1.dpR12<- (-1)*rep(c(1/pR1^2,1/(1-pR1)^2),4)
    d2h1.dpR22<- (-1)*rep(c(1/pR2^2,1/(1-pR2)^2),4)
    if (mapfun=="Haldane"){
      dpR1.dpR2 <- (1-2*pR2)*rno^2 /((1-2*pR1)*(1-rno)^2)
      d2pR1.dpR22<- (-2)*rno^2 /((1-2*pR1)*(1-rno)^2)
      dh1.dpR1<-rep(c(1/pR1,-1/(1-pR1)),4)
      tempi<- (d2h1.dpR12 * (dpR1.dpR2)^2 + dh1.dpR1 * d2pR1.dpR22)

      d2lnw[gg$NN00]<- matlist(d2lnw[gg$NN00], "dpR2",m$kgen,tempi)
      d2lnw[gg$NN0o]<- matlist(d2lnw[gg$NN0o], "dpR2",m$kgen,d2h1.dpR22)
      d2lnw[gg$Nn00]<- matlist(d2lnw[gg$Nn00], "dpR2",m$kgen,rev(d2h1.dpR22))
      d2lnw[gg$Nn0o]<- matlist(d2lnw[gg$Nn0o], "dpR2",m$kgen,rev(tempi))
    }
    else{
      d2lnw[gg$NN00]<- matlist(d2lnw[gg$NN00], "dpR1",m$kgen,d2h1.dpR12)
      d2lnw[gg$NN0o]<- matlist(d2lnw[gg$NN0o], "dpR2",m$kgen,d2h1.dpR22)
      d2lnw[gg$Nn00]<- matlist(d2lnw[gg$Nn00], "dpR2",m$kgen,rev(d2h1.dpR22))
      d2lnw[gg$Nn0o]<- matlist(d2lnw[gg$Nn0o], "dpR1",m$kgen,rev(d2h1.dpR12))
    }
  }
}
else{
  gg<-index.genot(cross="B1",N="N",genot2=genot2)
  d2lnw[gg$NN]<- matlist(d2lnw[gg$NN], "dpR",m$kgen,(-1)*rep(c(1/pR^2,1/(1-pR)^2),4))
  d2lnw[gg$Nn]<- matlist(d2lnw[gg$Nn], "dpR",m$kgen,rev((-1)*rep(c(1/pR^2,1/(1-pR)^2),4)))
}
}
if(length(m$Qfit)>0) {
  gg<-index.genot(cross="F2",M="M",N="N",genot2=genot2)
  d2h1.dpQ12<- (-1)*rep(c(1/pQ1^2,1/pQ1^2,1/(1-pQ1)^2,1/(1-pQ1)^2),2)
  d2h1.dpQ22<- (-1)*rep(c(1/pQ2^2,1/pQ2^2,1/(1-pQ2)^2, 1/(1-pQ2)^2),2)
  if (mapfun=="Haldane"){
    dpQ1.dpQ2 <- (1-2*pQ2)*rmn^2 /((1-2*pQ1)*(1-rmn)^2)
    d2pQ1.dpQ22<- (-2)*rmn^2 /((1-2*pQ1)*(1-rmn)^2)
    dh1.dpQ1<-rep(c(1/pQ1,1/pQ1,-1/(1-pQ1),-1/(1-pQ1)), 2 )
    tempi<- ( d2h1.dpQ12 * (dpQ1.dpQ2)^2+ dh1.dpQ1 * d2pQ1.dpQ22)

```

```

d2lnw[gg$MMNN]<- matlist(d2lnw[gg$MMNN], "dpQ2", m$kgen, tempi)
d2lnw[gg$MMNn]<- matlist(d2lnw[gg$MMNn], "dpQ2", m$kgen, d2h1.dpQ22)
d2lnw[gg$MmNN]<- matlist(d2lnw[gg$MmNN], "dpQ2", m$kgen, rev(d2h1.dpQ22))
d2lnw[gg$MmNn]<- matlist(d2lnw[gg$MmNn], "dpQ2", m$kgen, rev(tempi))
}
else{
d2lnw[gg$MMNN]<- matlist(d2lnw[gg$MMNN], "dpQ1", m$kgen, d2h1.dpQ12)
d2lnw[gg$MMNn]<- matlist(d2lnw[gg$MMNn], "dpQ2", m$kgen, d2h1.dpQ22)
d2lnw[gg$MmNN]<- matlist(d2lnw[gg$MmNN], "dpQ2", m$kgen, rev(d2h1.dpQ22))
d2lnw[gg$MmNn]<- matlist(d2lnw[gg$MmNn], "dpQ1", m$kgen, rev(d2h1.dpQ12))
}
}
d2lnw[markerg]
}

#-----
# D2w.Dphi2()
#-----
D2w.Dphi2<- function(cross,...){
  hessian<-switch(as.character(cross),
    B1=D2w.Dphi2.b1(...),
    B2=D2w.Dphi2.b1(...),
    F2=D2w.Dphi2.f2(...))
  hessian
}

#-----
# Dw.Dphi()
#-----
Dw.Dphi<- function(cross,...){
  derivative1<-switch(as.character(cross),
    B1=Dw.Dphi.b1(...),
    B2=Dw.Dphi.b1(...),
    F2=Dw.Dphi.f2(...))
  derivative1
}

#-----
# df2h1()
#-----
df2h1<-function(rkm,pL1,pL2,dpL1.dpL2,repv,h,repv2=NULL){
  df2<-switch(h,
    h1=c(2/pL1, 1/pL1-1/(1-pL1), -2/(1-pL1))*dpL1.dpL2,
    h2=c(1/pL1*dpL1.dpL2+1/pL2, ((1-2*pL1)+(1-2*pL2)*dpL1.dpL2)/(pL2*(1-pL1)+pL1*(1-pL2)),
      -1/(1-pL1)*dpL1.dpL2-1/(1-pL2) ),

```

```

h3=c(2/pL2, 1/pL2-1/(1-pL2), -2/(1-pL2)),
h4=c(1/pL1*dpL1.dpL2 -1/(1-pL2), ((2*pL1-1)+(2*pL2-1)*dpL1.dpL2)/(1-pL1-pL2+2*pL1*pL2),
1/pL2-1/(1-pL1)*dpL1.dpL2),
h5=c(((1-rkm)^2*(1-2*pL1)*dpL1.dpL2 + rkm^2*(1-2*pL2))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2)),
((1-rkm)^2*(-2+4*pL1)*dpL1.dpL2 + rkm^2*(-2+4*pL2))/
((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2))),
((1-rkm)^2*(1-2*pL1)*dpL1.dpL2 + rkm^2*(1-2*pL2))/
((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2)))
d.dpL2<-rep(df2,repv)
if (!is.null(repv2))
d.dpL2<-rep(d.dpL2,repv2)
d.dpL2
}

#-----
# df2g1()
#-----
df2g1<-function(rkm,pL1,pL2,repv,h,repv2=NULL){
df2<-switch(h,
g1.1=c(2/pL1, 1/pL1-1/(1-pL1), -2/(1-pL1)),
g2.1= c(1/pL1, (1-2*pL2)/(pL2*(1-pL1)+pL1*(1-pL2)), -1/(1-pL1)),
g2.2= c(1/pL2, (1-2*pL1)/(pL2*(1-pL1)+pL1*(1-pL2)), -1/(1-pL2)),
g3.2= c(2/pL2, 1/pL2-1/(1-pL2), -2/(1-pL2)),
g4.1= c(1/pL1, (-1+2*pL2)/(1-pL1-pL2+2*pL1*pL2), -1/(1-pL1)),
g4.2= c(-1/(1-pL2), (-1+2*pL1)/(1-pL1-pL2+2*pL1*pL2), 1/pL2),
g5.1= c(((1-rkm)^2*(1-2*pL1))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2)),
((1-rkm)^2*(-2+4*pL1))/((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2))),
((1-rkm)^2*(1-2*pL1))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))),
g5.2= c(( rkm^2*(1-2*pL2))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2)),
( rkm^2*(-2+4*pL2))/((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2))),
( rkm^2*(1-2*pL2))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))))
val<-rep(df2,repv)
if (!is.null(repv2))
val<-rep(val,repv2)
val
}

#-----
# Dw.Dphi.f2()
#-----
Dw.Dphi.f2<-function(mapfun,chosen.model,probs,markerg,genot,genot2,nqgen,m){
pQ1<-m$pQ1
pQ2<-m$pQ2
pL1<-m$pL1
pL2<-m$pL2

```

```

pR1<-m$pR1
pR2<-m$pR2
pL<-m$pL
pR<-m$pR
mconfig<-genot2$mconfig
rkm<-genot2$rkm
rmn<-genot2$rmn
rno<-genot2$rno

dfit.names<-c("dpL","dpR","dpL1","dpL2","dpR1","dpR2","dpQ1","dpQ2")
names(dfit.names)<-c("pL","pR","pL1","pL2","pR1","pR2","pQ1","pQ2")
dfit.names<-dfit.names[names(probs)]
mat<-matrix(0,nrow=nqgen,ncol=length(dfit.names))
dimnames(mat)<-list(NULL,dfit.names)
d1lnw<-lapply(1:length(genot),function(h,x){x},mat)
names(d1lnw)<-genot

matlist<-function(mylist,dqtl,index.qtl,val){
  lapply(mylist,function(x,dqtl,index.qtl,val){
    x[,dqtl]<-val[index.qtl]; x,
    dqtl,index.qtl,val)
  }
}

if(length(m$Lfit)>0) {
if (mconfig["K"]==1){
  gg<-index.genot(cross="F2",M="K",N="M",genot2=genot2)
if (mapfun=="Haldane"){
  dpL1.dpL2 <- (1-2*pL2)*rkm^2 /((1-2*pL1)*(1-rkm)^2)
  d1lnw[gg$KKMM]<-matlist(d1lnw[gg$KKMM],"dpL2",m$kgen,df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h1"))
  d1lnw[gg$KKMm]<-matlist(d1lnw[gg$KKMm],"dpL2",m$kgen,df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h2"))
  d1lnw[gg$KKmm]<-matlist(d1lnw[gg$KKmm],"dpL2",m$kgen,df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h3"))
  d1lnw[gg$KkMM]<-matlist(d1lnw[gg$KkMM],"dpL2",m$kgen,df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h4"))
  d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm],"dpL2",m$kgen,df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h5"))
  d1lnw[gg$Kkmm]<-matlist(d1lnw[gg$Kkmm],"dpL2",m$kgen,rev(df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h4")))
  d1lnw[gg$kkMM]<-matlist(d1lnw[gg$kkMM],"dpL2",m$kgen,rev(df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h3")))
  d1lnw[gg$kkMm]<-matlist(d1lnw[gg$kkMm],"dpL2",m$kgen,rev(df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h2")))
  d1lnw[gg$kkmm]<-matlist(d1lnw[gg$kkmm],"dpL2",m$kgen,rev(df2h1(rkm,pL1,pL2,dpL1.dpL2,c(9,9,9),"h1")))
}
}
else{
  d1lnw[gg$KKMM]<-matlist(d1lnw[gg$KKMM],"dpL1",m$kgen,df2g1(rkm,pL1,pL2,c(9,9,9),"g1.1"))
  d1lnw[gg$KKMm]<-matlist(d1lnw[gg$KKMm],"dpL1",m$kgen,df2g1(rkm,pL1,pL2,c(9,9,9),"g2.1"))
  d1lnw[gg$KKmm]<-matlist(d1lnw[gg$KKmm],"dpL2",m$kgen,df2g1(rkm,pL1,pL2,c(9,9,9),"g2.2"))
  d1lnw[gg$KkMM]<-matlist(d1lnw[gg$KkMM],"dpL2",m$kgen,df2g1(rkm,pL1,pL2,c(9,9,9),"g3.2"))
  d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm],"dpL1",m$kgen,df2g1(rkm,pL1,pL2,c(9,9,9),"g4.1"))
  d1lnw[gg$Kkmm]<-matlist(d1lnw[gg$Kkmm],"dpL2",m$kgen,df2g1(rkm,pL1,pL2,c(9,9,9),"g4.2"))
}
}

```

```

d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm], "dpL1", m$kgen, df2g1(rkm, pL1, pL2, c(9,9,9), "g5.1"))
d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm], "dpL2", m$kgen, df2g1(rkm, pL1, pL2, c(9,9,9), "g5.2"))

d1lnw[gg$Kkmm]<-matlist(d1lnw[gg$Kkmm], "dpL1", m$kgen, rev(df2g1(rkm, pL1, pL2, c(9,9,9), "g4.1")))
d1lnw[gg$Kkmm]<-matlist(d1lnw[gg$Kkmm], "dpL2", m$kgen, rev(df2g1(rkm, pL1, pL2, c(9,9,9), "g4.2")))
d1lnw[gg$kkMM]<-matlist(d1lnw[gg$kkMM], "dpL1", m$kgen, rev(df2g1(rkm, pL1, pL2, c(9,9,9), "g3.2")))
d1lnw[gg$kkMm]<-matlist(d1lnw[gg$kkMm], "dpL1", m$kgen, rev(df2g1(rkm, pL1, pL2, c(9,9,9), "g2.1")))
d1lnw[gg$kkMm]<-matlist(d1lnw[gg$kkMm], "dpL2", m$kgen, rev(df2g1(rkm, pL1, pL2, c(9,9,9), "g2.2")))
d1lnw[gg$kkmm]<-matlist(d1lnw[gg$kkmm], "dpL1", m$kgen, rev(df2g1(rkm, pL1, pL2, c(9,9,9), "g1.1")))
}
}
else{
  gg<-index.genot(cross="F2", M="M", genot2=genot2)
  d1lnw[gg$MM]<- matlist(d1lnw[gg$MM], "dpL", m$kgen, rep(c(2/pL, 1/pL-1/(1-pL), -2/(1-pL)), c(9,9,9)))
  d1lnw[gg$Mm]<- matlist(d1lnw[gg$Mm], "dpL", m$kgen, rep(c(1/pL-1/(1-pL), (-2+4*pL)/(pL^2+(1-pL)^2),
    1/pL-1/(1-pL)), c(9,9,9)))
  d1lnw[gg$mm]<- matlist(d1lnw[gg$mm], "dpL", m$kgen,
    rev(rep(c(2/pL, 1/pL-1/(1-pL), -2/(1-pL)), c(9,9,9))))
}
}

if(length(m$Rfit)>0) {
  if (mconfig["0"]==1){
    gg<-index.genot(cross="F2", M="N", N="0", genot2=genot2)
    if (mapfun=="Haldane"){
      dpR1.dpR2 <- (1-2*pR2)*rno^2 /((1-2*pR1)*(1-rno)^2)

      d1lnw[gg$NN00]<-matlist(d1lnw[gg$NN00], "dpR2", m$kgen, df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h1"))
      d1lnw[gg$NN0o]<-matlist(d1lnw[gg$NN0o], "dpR2", m$kgen, df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h2"))
      d1lnw[gg$NNoo]<-matlist(d1lnw[gg$NNoo], "dpR2", m$kgen, df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h3"))
      d1lnw[gg$Nn00]<-matlist(d1lnw[gg$Nn00], "dpR2", m$kgen, df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h4"))
      d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn0o], "dpR2", m$kgen, df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h5"))
      d1lnw[gg$Nnoo]<-matlist(d1lnw[gg$Nnoo], "dpR2", m$kgen, rev(df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h4")))
      d1lnw[gg$nn00]<-matlist(d1lnw[gg$nn00], "dpR2", m$kgen, rev(df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h3")))
      d1lnw[gg$nn0o]<-matlist(d1lnw[gg$nn0o], "dpR2", m$kgen, rev(df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h2")))
      d1lnw[gg$nnoo]<-matlist(d1lnw[gg$nnoo], "dpR2", m$kgen, rev(df2h1(rno, pR1, pR2, dpR1.dpR2, 9, "h1"))) }
    else{
      d1lnw[gg$NN00]<-matlist(d1lnw[gg$NN00], "dpR1", m$kgen, df2g1(rno, pR1, pR2, 9, "g1.1"))
      d1lnw[gg$NN0o]<-matlist(d1lnw[gg$NN0o], "dpR1", m$kgen, df2g1(rno, pR1, pR2, 9, "g2.1"))
      d1lnw[gg$NN0o]<-matlist(d1lnw[gg$NN0o], "dpR2", m$kgen, df2g1(rno, pR1, pR2, 9, "g2.2"))
      d1lnw[gg$NNoo]<-matlist(d1lnw[gg$NNoo], "dpR2", m$kgen, df2g1(rno, pR1, pR2, 9, "g3.2"))
      d1lnw[gg$Nn00]<-matlist(d1lnw[gg$Nn00], "dpR1", m$kgen, df2g1(rno, pR1, pR2, 9, "g4.1"))
      d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn0o], "dpR2", m$kgen, df2g1(rno, pR1, pR2, 9, "g4.2"))
      d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn0o], "dpR1", m$kgen, df2g1(rno, pR1, pR2, 9, "g5.1"))
      d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn0o], "dpR2", m$kgen, df2g1(rno, pR1, pR2, 9, "g5.2"))
    }
  }
}

```

```

d1lnw[gg$Nnoo]<-matlist(d1lnw[gg$Nnoo], "dpR1", m$kgen, rev(df2g1(rno, pR1, pR2, 9, "g4.1")))
d1lnw[gg$Nnoo]<-matlist(d1lnw[gg$Nnoo], "dpR2", m$kgen, rev(df2g1(rno, pR1, pR2, 9, "g4.2")))
d1lnw[gg$nn00]<-matlist(d1lnw[gg$nn00], "dpR2", m$kgen, rev(df2g1(rno, pR1, pR2, 9, "g3.2")))
d1lnw[gg$nn0o]<-matlist(d1lnw[gg$nn0o], "dpR1", m$kgen, rev(df2g1(rno, pR1, pR2, 9, "g2.1")))
d1lnw[gg$nn0o]<-matlist(d1lnw[gg$nn0o], "dpR2", m$kgen, rev(df2g1(rno, pR1, pR2, 9, "g2.2")))
d1lnw[gg$nnoo]<-matlist(d1lnw[gg$nnoo], "dpR1", m$kgen, rev(df2g1(rno, pR1, pR2, 9, "g1.1")))
}
}
else{
  gg<-index.genot(cross="F2", N="N", genot2=genot2)
  d1lnw[gg$NN]<- matlist(d1lnw[gg$NN], "dpR", m$kgen, rep(c(2/pR, 1/pR-1/(1-pR), -2/(1-pR)), 9))
  d1lnw[gg$Nn]<- matlist(d1lnw[gg$Nn], "dpR", m$kgen,
    rep(c(1/pR-1/(1-pR), (-2+4*pR)/(pR^2+(1-pR)^2), 1/pR-1/(1-pR)), 9))
  d1lnw[gg$nn]<- matlist(d1lnw[gg$nn], "dpR", m$kgen, rev(rep(c(2/pR, 1/pR-1/(1-pR), -2/(1-pR)), 9)))
}
}
if(length(m$Qfit)>0) {
  gg<-index.genot(cross="F2", M="M", N="N", genot2=genot2)
  if (mapfun=="Haldane"){
    dpQ1.dpQ2 <- (1-2*pQ2)*rmn^2 /((1-2*pQ1)*(1-rmn)^2)
    d1lnw[gg$MMNN]<-matlist(d1lnw[gg$MMNN], "dpQ2", m$kgen, df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h1", 3))
    d1lnw[gg$MMNn]<-matlist(d1lnw[gg$MMNn], "dpQ2", m$kgen, df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h2", 3))
    d1lnw[gg$MMnn]<-matlist(d1lnw[gg$MMnn], "dpQ2", m$kgen, df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h3", 3))
    d1lnw[gg$MmNN]<-matlist(d1lnw[gg$MmNN], "dpQ2", m$kgen, df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h4", 3))
    d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ2", m$kgen, df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h5", 3))
    d1lnw[gg$Mmnn]<-matlist(d1lnw[gg$Mmnn], "dpQ2", m$kgen,
      rev(df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h4", 3)))
    d1lnw[gg$mmNN]<-matlist(d1lnw[gg$mmNN], "dpQ2", m$kgen,
      rev(df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h3", 3)))
    d1lnw[gg$mmNn]<-matlist(d1lnw[gg$mmNn], "dpQ2", m$kgen,
      rev(df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h2", 3)))
    d1lnw[gg$mmnn]<-matlist(d1lnw[gg$mmnn], "dpQ2", m$kgen,
      rev(df2h1(rmn, pQ1, pQ2, dpQ1.dpQ2, c(3,3,3), "h1", 3)))
  }
  else{
    d1lnw[gg$MMNN]<-matlist(d1lnw[gg$MMNN], "dpQ1", m$kgen, df2g1(rmn, pQ1, pQ2, c(3,3,3), "g1.1", 3))
    d1lnw[gg$MMNn]<-matlist(d1lnw[gg$MMNn], "dpQ1", m$kgen, df2g1(rmn, pQ1, pQ2, c(3,3,3), "g2.1", 3))
    d1lnw[gg$MMnn]<-matlist(d1lnw[gg$MMnn], "dpQ2", m$kgen, df2g1(rmn, pQ1, pQ2, c(3,3,3), "g2.2", 3))
    d1lnw[gg$MMnn]<-matlist(d1lnw[gg$MMnn], "dpQ2", m$kgen, df2g1(rmn, pQ1, pQ2, c(3,3,3), "g3.2", 3))
    d1lnw[gg$MmNN]<-matlist(d1lnw[gg$MmNN], "dpQ1", m$kgen, df2g1(rmn, pQ1, pQ2, c(3,3,3), "g4.1", 3))
    d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ2", m$kgen, df2g1(rmn, pQ1, pQ2, c(3,3,3), "g4.2", 3))
    d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ1", m$kgen, df2g1(rmn, pQ1, pQ2, c(3,3,3), "g5.1", 3))
    d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ2", m$kgen, df2g1(rmn, pQ1, pQ2, c(3,3,3), "g5.2", 3))
  }
}

```



```

d1lnw[gg$Mmnn]<-matlist(d1lnw[gg$Mmnn], "dpQ1", m$kgen, rev(df2g1(rmn,pQ1,pQ2,c(3,3,3), "g4.1", 3)))
d1lnw[gg$Mmnn]<-matlist(d1lnw[gg$Mmnn], "dpQ2", m$kgen, rev(df2g1(rmn,pQ1,pQ2,c(3,3,3), "g4.2", 3)))
d1lnw[gg$mmNN]<-matlist(d1lnw[gg$mmNN], "dpQ2", m$kgen, rev(df2g1(rmn,pQ1,pQ2,c(3,3,3), "g3.2", 3)))
d1lnw[gg$mmNn]<-matlist(d1lnw[gg$mmNn], "dpQ1", m$kgen, rev(df2g1(rmn,pQ1,pQ2,c(3,3,3), "g2.1", 3)))
d1lnw[gg$mmNn]<-matlist(d1lnw[gg$mmNn], "dpQ2", m$kgen, rev(df2g1(rmn,pQ1,pQ2,c(3,3,3), "g2.2", 3)))
d1lnw[gg$mmnn]<-matlist(d1lnw[gg$mmnn], "dpQ1", m$kgen, rev(df2g1(rmn,pQ1,pQ2,c(3,3,3), "g1.1", 3)))
}
}
d1lnw[markerg]
}

```

```

#-----
# D2w.Dphi2.f2()
#-----
D2w.Dphi2.f2<-function(mapfun,chosen.model,probs,markerg,genot,genot2,nqgen,m){
  pQ1<-m$pQ1
  pQ2<-m$pQ2
  pL1<-m$pL1
  pL2<-m$pL2
  pR1<-m$pR1
  pR2<-m$pR2
  pL<-m$pL
  pR<-m$pR
  mconfig<-genot2$mconfig
  rkm<-genot2$rkm
  rmn<-genot2$rmn
  rno<-genot2$rno

  dfit.names<-c("dpL", "dpR", "dpL1", "dpL2", "dpR1", "dpR2", "dpQ1", "dpQ2")
  names(dfit.names)<-c("pL", "pR", "pL1", "pL2", "pR1", "pR2", "pQ1", "pQ2")
  dfit.names<-dfit.names[names(probs)]
  mat<-matrix(0,nrow=length(dfit.names),ncol=nqgen)
  dimnames(mat)<-list(dfit.names,NULL)
  d2h1<-as.list(NULL)
  d2h1<-lapply(1:length(dfit.names),function(h,x){x},mat)
  names(d2h1)<-dfit.names
  d2lnw<-lapply(1:length(genot),function(h,x){x},d2h1)
  names(d2lnw)<-genot

  #indexing a three-level nested list using the fact that dpR dpL=0 etc
  matlist<-function(mylist,dp1,dp2,index.qtl,val){
    lapply(mylist,function(x,dp1,dp2,index.qtl,val){

```

```

      x[[dp1]][dp2,]<-val[index.qtl]; x},
      dp1,dp2,index.qtl,val)
    }

if(length(m$Lfit)>0) {
  if (mconfig["K"]==1){
    gg<-index.genot(cross="F2",M="K",N="M",genot2=genot2)

    if (mapfun=="Haldane"){
      dpL1.dpL2 <- (1-2*pL2)*rkm^2 /((1-2*pL1)*(1-rkm)^2)
      d2pL1.dpL22<- (-2)*rkm^2 /((1-2*pL1)*(1-rkm)^2)

      d2lnw[gg$KKMM]<-matlist(d2lnw[gg$KKMM], "dpL2", "dpL2", m$kgen,
                              d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h1"))
      d2lnw[gg$KKMm]<-matlist(d2lnw[gg$KKMm], "dpL2", "dpL2", m$kgen,
                              d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h2"))
      d2lnw[gg$KKmm]<-matlist(d2lnw[gg$KKmm], "dpL2", "dpL2", m$kgen,
                              d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h3"))
      d2lnw[gg$KkMM]<-matlist(d2lnw[gg$KkMM], "dpL2", "dpL2", m$kgen,
                              d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h4"))
      d2lnw[gg$KkMm]<-matlist(d2lnw[gg$KkMm], "dpL2", "dpL2", m$kgen,
                              d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h5"))
      d2lnw[gg$Kkmm]<-matlist(d2lnw[gg$Kkmm], "dpL2", "dpL2", m$kgen,
                              rev(d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h4")))
      d2lnw[gg$kkMM]<-matlist(d2lnw[gg$kkMM], "dpL2", "dpL2", m$kgen,
                              rev(d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h3")))
      d2lnw[gg$kkMm]<-matlist(d2lnw[gg$kkMm], "dpL2", "dpL2", m$kgen,
                              rev(d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h2")))
      d2lnw[gg$kkmm]<-matlist(d2lnw[gg$kkmm], "dpL2", "dpL2", m$kgen,
                              rev(d2f2h1(rkm,pL1,pL2,dpL1.dpL2,d2pL1.dpL22,c(9,9,9), "h1")))
    }
  } else{
    d1lnw[gg$KKMM]<-matlist(d1lnw[gg$KKMM], "dpL1", "dpL1", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p1.g1.1"))
    d1lnw[gg$KKMm]<-matlist(d1lnw[gg$KKMm], "dpL1", "dpL1", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p1.g2.1"))
    d1lnw[gg$KKMm]<-matlist(d1lnw[gg$KKMm], "dpL2", "dpL1", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p2.g2.1"))
    d1lnw[gg$KKMm]<-matlist(d1lnw[gg$KKMm], "dpL1", "dpL2", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p1.g2.2"))
    d1lnw[gg$KKMm]<-matlist(d1lnw[gg$KKMm], "dpL2", "dpL2", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p2.g2.2"))
    d1lnw[gg$KKmm]<-matlist(d1lnw[gg$KKmm], "dpL2", "dpL2", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p2.g3.2"))
    d1lnw[gg$KkMM]<-matlist(d1lnw[gg$KkMM], "dpL1", "dpL1", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p1.g4.1"))
    d1lnw[gg$KkMM]<-matlist(d1lnw[gg$KkMM], "dpL2", "dpL1", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p2.g4.1"))
    d1lnw[gg$KkMM]<-matlist(d1lnw[gg$KkMM], "dpL1", "dpL2", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p1.g4.2"))
    d1lnw[gg$KkMM]<-matlist(d1lnw[gg$KkMM], "dpL2", "dpL2", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p1.g4.2"))
    d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm], "dpL1", "dpL1", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p1.g5.1"))
    d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm], "dpL2", "dpL1", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p2.g5.1"))
    d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm], "dpL1", "dpL2", m$kgen, d2f2g1(rkm,pL1,pL2,c(9,9,9), "p1.g5.2"))
  }
}

```

```

d1lnw[gg$KkMm]<-matlist(d1lnw[gg$KkMm], "dpL2", "dpL2", m$kgen, d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p2.g5.2"))

d1lnw[gg$Kkmm]<-matlist(d1lnw[gg$Kkmm], "dpL1", "dpL1", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p1.g4.1")))
d1lnw[gg$Kkmm]<-matlist(d1lnw[gg$Kkmm], "dpL2", "dpL1", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p2.g4.1")))
d1lnw[gg$Kkmm]<-matlist(d1lnw[gg$Kkmm], "dpL1", "dpL2", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p1.g4.2")))
d1lnw[gg$Kkmm]<-matlist(d1lnw[gg$Kkmm], "dpL2", "dpL2", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p2.g4.2")))
d1lnw[gg$kkMM]<-matlist(d1lnw[gg$kkMM], "dpL2", "dpL2", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p2.g3.2")))
d1lnw[gg$kkMm]<-matlist(d1lnw[gg$kkMm], "dpL1", "dpL1", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p1.g2.1")))
d1lnw[gg$kkMm]<-matlist(d1lnw[gg$kkMm], "dpL2", "dpL1", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p2.g2.1")))
d1lnw[gg$kkMm]<-matlist(d1lnw[gg$kkMm], "dpL1", "dpL2", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p1.g2.2")))
d1lnw[gg$kkMm]<-matlist(d1lnw[gg$kkMm], "dpL2", "dpL2", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p2.g2.2")))
d1lnw[gg$kkmm]<-matlist(d1lnw[gg$kkmm], "dpL1", "dpL1", m$kgen,
  rev(d2f2g1(rkm, pL1, pL2, c(9, 9, 9), "p1.g1.1")))
}
}
else{
  gg<-index.genot(cross="F2", M="M", genot2=genot2)
  d2lnw[gg$MM]<- matlist(d2lnw[gg$MM], "dpL", "dpL", m$kgen,
    rep(c(-2/pL^2, -1/pL^2-1/(1-pL)^2, -2/(1-pL)^2), c(9, 9, 9)))
  d2lnw[gg$Mm]<- matlist(d2lnw[gg$Mm], "dpL", "dpL", m$kgen, rep(c(-1/pL^2-1/(1-pL)^2,
    (4*(pL^2+(1-pL)^2)-(-2+4*pL)^2)/(pL^2+(1-pL)^2), -1/pL^2-1/(1-pL)^2), c(9, 9, 9)))
  d2lnw[gg$mm]<- matlist(d2lnw[gg$mm], "dpL", "dpL", m$kgen,
    rev(rep(c(-2/pL^2, -1/pL^2-1/(1-pL)^2, -2/(1-pL)^2), c(9, 9, 9))))
}
}
if(length(m$Rfit)>0) {
  if (mconfig["0"]==1){
    gg<-index.genot(cross="F2", M="N", N="0", genot2=genot2)

    if (mapfun=="Haldane"){
      dpR1.dpR2 <- (1-2*pR2)*rno^2 /((1-2*pR1)*(1-rno)^2)
      d2pR1.dpR22<- (-2)*rno^2 /((1-2*pR1)*(1-rno)^2)
      d2lnw[gg$NN00]<-matlist(d2lnw[gg$NN00], "dpR2", "dpR2", m$kgen,
        d2f2h1(rno, pR1, pR2, dpR1.dpR2, d2pR1.dpR22, 9, "h1"))
      d2lnw[gg$NN0o]<-matlist(d2lnw[gg$NN0o], "dpR2", "dpR2", m$kgen,

```

```

        d2f2h1(rno,pR1,pR2,dpR1.dpR2,d2pR1.dpR22,9,"h2"))
d2lnw[gg$NNoo]<-matlist(d2lnw[gg$NNoo],"dpR2","dpR2",m$kgen,
        d2f2h1(rno,pR1,pR2,dpR1.dpR2,d2pR1.dpR22,9,"h3"))
d2lnw[gg$Nn00]<-matlist(d2lnw[gg$Nn00],"dpR2","dpR2",m$kgen,
        d2f2h1(rno,pR1,pR2,dpR1.dpR2,d2pR1.dpR22,9,"h4"))
d2lnw[gg$Nn0o]<-matlist(d2lnw[gg$Nn0o],"dpR2","dpR2",m$kgen,
        d2f2h1(rno,pR1,pR2,dpR1.dpR2,d2pR1.dpR22,9,"h5"))
d2lnw[gg$Nnoo]<-matlist(d2lnw[gg$Nnoo],"dpR2","dpR2",m$kgen,
        rev(d2f2h1(rno,pR1,pR2,dpR1.dpR2,d2pR1.dpR22,9,"h4"))))
d2lnw[gg$nn00]<-matlist(d2lnw[gg$nn00],"dpR2","dpR2",m$kgen,
        rev(d2f2h1(rno,pR1,pR2,dpR1.dpR2,d2pR1.dpR22,9,"h3"))))
d2lnw[gg$nn0o]<-matlist(d2lnw[gg$nn0o],"dpR2","dpR2",m$kgen,
        rev(d2f2h1(rno,pR1,pR2,dpR1.dpR2,d2pR1.dpR22,9,"h2"))))
d2lnw[gg$nnoo]<-matlist(d2lnw[gg$nnoo],"dpR2","dpR2",m$kgen,
        rev(d2f2h1(rno,pR1,pR2,dpR1.dpR2,d2pR1.dpR22,9,"h1")))#h1rev
}
else{
    d1lnw[gg$NN00]<-matlist(d1lnw[gg$NN00],"dpR1","dpR1",m$kgen,d2f2g1(rno,pR1,pR2,9,"p1.g1.1"))
    d1lnw[gg$NN0o]<-matlist(d1lnw[gg$NN0o],"dpR1","dpR1",m$kgen,d2f2g1(rno,pR1,pR2,9,"p1.g2.1"))
    d1lnw[gg$NN0o]<-matlist(d1lnw[gg$NN0o],"dpR2","dpR1",m$kgen,d2f2g1(rno,pR1,pR2,9,"p2.g2.1"))
    d1lnw[gg$NN0o]<-matlist(d1lnw[gg$NN0o],"dpR1","dpR2",m$kgen,d2f2g1(rno,pR1,pR2,9,"p1.g2.2"))
    d1lnw[gg$NN0o]<-matlist(d1lnw[gg$NN0o],"dpR2","dpR2",m$kgen,d2f2g1(rno,pR1,pR2,9,"p2.g2.2"))
    d1lnw[gg$Nnoo]<-matlist(d1lnw[gg$Nnoo],"dpR2","dpR2",m$kgen,d2f2g1(rno,pR1,pR2,9,"p2.g3.2"))
    d1lnw[gg$Nn00]<-matlist(d1lnw[gg$Nn00],"dpR1","dpR1",m$kgen,d2f2g1(rno,pR1,pR2,9,"p1.g4.1"))
    d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn00],"dpR2","dpR1",m$kgen,d2f2g1(rno,pR1,pR2,9,"p2.g4.1"))
    d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn00],"dpR1","dpR2",m$kgen,d2f2g1(rno,pR1,pR2,9,"p1.g4.2"))
    d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn00],"dpR2","dpR2",m$kgen,d2f2g1(rno,pR1,pR2,9,"p2.g4.2"))
    d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn0o],"dpR1","dpR1",m$kgen,d2f2g1(rno,pR1,pR2,9,"p1.g5.1"))
    d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn0o],"dpR2","dpR1",m$kgen,d2f2g1(rno,pR1,pR2,9,"p2.g5.1"))
    d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn0o],"dpR1","dpR2",m$kgen,d2f2g1(rno,pR1,pR2,9,"p1.g5.2"))
    d1lnw[gg$Nn0o]<-matlist(d1lnw[gg$Nn0o],"dpR2","dpR2",m$kgen,d2f2g1(rno,pR1,pR2,9,"p2.g5.2"))

    d1lnw[gg$Nnoo]<-matlist(d1lnw[gg$Nnoo],"dpR1","dpR1",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p1.g4.1"))))
    d1lnw[gg$Nnoo]<-matlist(d1lnw[gg$Nnoo],"dpR2","dpR1",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p2.g4.1"))))
    d1lnw[gg$Nnoo]<-matlist(d1lnw[gg$Nnoo],"dpR1","dpR2",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p1.g4.2"))))
    d1lnw[gg$Nnoo]<-matlist(d1lnw[gg$Nnoo],"dpR2","dpR2",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p2.g4.2"))))
    d1lnw[gg$nn00]<-matlist(d1lnw[gg$nn00],"dpR2","dpR2",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p2.g3.2"))))
    d1lnw[gg$nn0o]<-matlist(d1lnw[gg$nn0o],"dpR1","dpR1",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p1.g2.1"))))
    d1lnw[gg$nn0o]<-matlist(d1lnw[gg$nn0o],"dpR2","dpR1",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p2.g2.1"))))
    d1lnw[gg$nn0o]<-matlist(d1lnw[gg$nn0o],"dpR1","dpR2",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p1.g2.2"))))
    d1lnw[gg$nn0o]<-matlist(d1lnw[gg$nn0o],"dpR2","dpR2",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p2.g2.2"))))
    d1lnw[gg$nnoo]<-matlist(d1lnw[gg$nnoo],"dpR1","dpR1",m$kgen,rev(d2f2g1(rno,pR1,pR2,9,"p1.g1.1"))))
}
}
else{

```

```

gg<-index.genot(cross="B1",N="N",genot2=genot2)
d2lnw[gg$NN]<- matlist(d2lnw[gg$NN], "dpR", "dpR", m$kgen,
                      rep(c(-2/pR^2, -1/pR^2-1/(1-pR)^2, -2/(1-pR)^2), 9))
d2lnw[gg$Nn]<- matlist(d2lnw[gg$Nn], "dpR", "dpR", m$kgen, rep(c(-1/pR^2-1/(1-pR)^2,
                      (4*(pR^2+(1-pR)^2)-(-2+4*pR)^2)/(pR^2+(1-pR)^2), -1/pR^2-1/(1-pR)^2), 9))
d2lnw[gg$nn]<- matlist(d2lnw[gg$nn], "dpR", "dpR", m$kgen,
                      rev(rep(c(-2/pR^2, -1/pR^2-1/(1-pR)^2, -2/(1-pR)^2), 9)))
}
}
if(length(m$Qfit)>0) {
  gg<-index.genot(cross="F2",M="M",N="N",genot2=genot2)

  if (mapfun=="Haldane"){
    dpQ1.dpQ2 <- (1-2*pQ2)*rmn^2 /((1-2*pQ1)*(1-rmn)^2)
    d2pQ1.dpQ22<- (-2)*rmn^2 /((1-2*pQ1)*(1-rmn)^2)
    d2lnw[gg$MMNN]<-matlist(d2lnw[gg$MMNN], "dpQ2", "dpQ2", m$kgen,
                          d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h1", 3))
    d2lnw[gg$MMNn]<-matlist(d2lnw[gg$MMNn], "dpQ2", "dpQ2", m$kgen,
                          d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h2", 3))
    d2lnw[gg$MMnn]<-matlist(d2lnw[gg$MMnn], "dpQ2", "dpQ2", m$kgen,
                          d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h3", 3))
    d2lnw[gg$MmNN]<-matlist(d2lnw[gg$MmNN], "dpQ2", "dpQ2", m$kgen,
                          d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h4", 3))
    d2lnw[gg$MmNn]<-matlist(d2lnw[gg$MmNn], "dpQ2", "dpQ2", m$kgen,
                          d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h5", 3))
    d2lnw[gg$Mmnn]<-matlist(d2lnw[gg$Mmnn], "dpQ2", "dpQ2", m$kgen,
                          rev(d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h4", 3)))
    d2lnw[gg$mmNN]<-matlist(d2lnw[gg$mmNN], "dpQ2", "dpQ2", m$kgen,
                          rev(d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h3", 3)))
    d2lnw[gg$mmNn]<-matlist(d2lnw[gg$mmNn], "dpQ2", "dpQ2", m$kgen,
                          rev(d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h2", 3)))
    d2lnw[gg$mmnn]<-matlist(d2lnw[gg$mmnn], "dpQ2", "dpQ2", m$kgen,
                          rev(d2f2h1(rmn,pQ1,pQ2,dpQ1.dpQ2,d2pQ1.dpQ22,c(3,3,3), "h1", 3)))
  }
  else{
    d1lnw[gg$MMNN]<-matlist(d1lnw[gg$MMNN], "dpQ1", "dpQ1", m$kgen,
                          d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g1.1", 3))
    d1lnw[gg$MMNn]<-matlist(d1lnw[gg$MMNn], "dpQ1", "dpQ1", m$kgen,
                          d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g2.1", 3))
    d1lnw[gg$MMnn]<-matlist(d1lnw[gg$MMnn], "dpQ2", "dpQ1", m$kgen,
                          d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g2.1", 3))
    d1lnw[gg$MMNn]<-matlist(d1lnw[gg$MMNn], "dpQ1", "dpQ2", m$kgen,
                          d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g2.2", 3))
    d1lnw[gg$MMnn]<-matlist(d1lnw[gg$MMnn], "dpQ2", "dpQ2", m$kgen,
                          d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g2.2", 3))
  }
}

```

```

d1lnw[gg$MMnn]<-matlist(d1lnw[gg$MMnn], "dpQ2", "dpQ2", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g3.2", 3))
d1lnw[gg$MmNN]<-matlist(d1lnw[gg$MmNN], "dpQ1", "dpQ1", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g4.1", 3))
d1lnw[gg$MmNN]<-matlist(d1lnw[gg$MmNN], "dpQ2", "dpQ1", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g4.1", 3))
d1lnw[gg$MmNN]<-matlist(d1lnw[gg$MmNN], "dpQ1", "dpQ2", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g4.2", 3))
d1lnw[gg$MmNN]<-matlist(d1lnw[gg$MmNN], "dpQ2", "dpQ2", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g4.2", 3))
d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ1", "dpQ1", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g5.1", 3))
d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ2", "dpQ1", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g5.1", 3))
d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ1", "dpQ2", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g5.2", 3))
d1lnw[gg$MmNn]<-matlist(d1lnw[gg$MmNn], "dpQ2", "dpQ2", m$kgen,
                        d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g5.2", 3))

d1lnw[gg$Mmnn]<-matlist(d1lnw[gg$Mmnn], "dpQ1", "dpQ1", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g4.1", 3)))
d1lnw[gg$Mmnn]<-matlist(d1lnw[gg$Mmnn], "dpQ2", "dpQ1", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g4.1", 3)))
d1lnw[gg$Mmnn]<-matlist(d1lnw[gg$Mmnn], "dpQ1", "dpQ2", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g4.2", 3)))
d1lnw[gg$Mmnn]<-matlist(d1lnw[gg$Mmnn], "dpQ2", "dpQ2", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g4.2", 3)))
d1lnw[gg$mmNN]<-matlist(d1lnw[gg$mmNN], "dpQ2", "dpQ2", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g3.2", 3)))
d1lnw[gg$mmNn]<-matlist(d1lnw[gg$mmNn], "dpQ1", "dpQ1", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g2.1", 3)))
d1lnw[gg$mmNn]<-matlist(d1lnw[gg$mmNn], "dpQ2", "dpQ1", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g2.1", 3)))
d1lnw[gg$mmNn]<-matlist(d1lnw[gg$mmNn], "dpQ1", "dpQ2", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g2.2", 3)))
d1lnw[gg$mmNn]<-matlist(d1lnw[gg$mmNn], "dpQ2", "dpQ2", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p2.g2.2", 3)))
d1lnw[gg$mmnn]<-matlist(d1lnw[gg$mmnn], "dpQ1", "dpQ1", m$kgen,
                        rev(d2f2g1(rmn,pQ1,pQ2,c(3,3,3), "p1.g1.1", 3)))
}
}
d2lnw[markerg]
}

```

#-----

```

# d2f2h1()
#-----
d2f2h1<-function(rkm,pL1,pL2,dpL1,dpL2,d2pL1,dpL22,repv,h,repv2=NULL){
  d2f2<-switch(h,
    h1=c(2/pL1, 1/pL1-1/(1-pL1), -2/(1-pL1))*d2pL1.dpL22
        +c(-2/pL1^2, -1/pL1^2-1/(1-pL1)^2, -2/(1-pL1)^2)*dpL1.dpL2^2,

    h2=c(1/pL1*d2pL1.dpL22 -1/pL1^2*dpL1.dpL2^2 -1/pL2^2,
        ((pL2*(1-pL1)+pL1*(1-pL2))*(-4*dpL1.dpL2+(1-2*pL2)*d2pL1.dpL22)
        -((1-2*pL1)+(1-2*pL2)*dpL1.dpL2)^2)/(pL2*(1-pL1)+pL1*(1-pL2))^2,
        -1/(1-pL1)*d2pL1.dpL22-1/(1-pL1)^2*dpL1.dpL2^2-1/(1-pL2)^2),

    h3=c(-2/pL2^2, -1/pL2^2-1/(1-pL2)^2, -2/(1-pL2)^2),

    h4=c(1/pL1*d2pL1.dpL22-1/pL1^2*dpL1.dpL2^2 -1/(1-pL2)^2,
        (((1-pL1-pL2+2*pL1*pL2)*(4*dpL1.dpL2+(2*pL2-1)*d2pL1.dpL22)
        -((2*pL1-1)+(2*pL2-1)*dpL1.dpL2)^2)/(1-pL1-pL2+2*pL1*pL2)^2),
        -1/pL2^2-1/(1-pL1)*d2pL1.dpL22-1/(1-pL1)^2*dpL1.dpL2^2),

    h5=c((((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))*((
        (1-rkm)^2*((1-2*pL1)*d2pL1.dpL22-2*dpL1.dpL2^2)-2*rkm^2)
        -((1-rkm)^2*(1-2*pL1)*dpL1.dpL2 + rkm^2*(1-2*pL2))^2 )/
        ((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2),

        (((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2)))*
        ((1-rkm)^2*((-2+4*pL1)*d2pL1.dpL22+4*dpL1.dpL2^2 )+ 4*rkm^2)-
        ((1-rkm)^2*(-2+4*pL1)*dpL1.dpL2 + rkm^2*(-2+4*pL2))^2)/
        ((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2)))^2,

        (((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))*((
        (1-rkm)^2*((1-2*pL1)*d2pL1.dpL22-2*dpL1.dpL2^2)-2*rkm^2)
        -((1-rkm)^2*(1-2*pL1)*dpL1.dpL2 + rkm^2*(1-2*pL2))^2 )/
        ((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2)
    ))
  d2.dpL2<-rep(d2f2,repv)
  if (!is.null(repv2))
    d2.dpL2<-rep(d2.dpL2,repv2)
  d2.dpL2
}

#-----
# d2f2g1()
#-----
d2f2g1<-function(rkm,pL1,pL2,repv,h,repv2=NULL){
  d2f2<-switch(h,

```

$$\begin{aligned}
p1.g1.1 &= c(-2/pL1^2, -1/pL1^2-1/(1-pL1)^2, -2/(1-pL1)^2), \\
p1.g2.1 &= c(-1/pL1^2, -(1-2*pL2)^2/(pL2*(1-pL1)+pL1*(1-pL2))^2, -1/(1-pL1)^2), \\
p2.g2.1 &= c(0, ((pL2*(1-pL1)+pL1*(1-pL2))*(-2)-(1-2*pL2)*(1-2*pL1))/(pL2*(1-pL1)+pL1*(1-pL2))^2, 0), \\
p1.g2.2 &= c(0, ((pL2*(1-pL1)+pL1*(1-pL2))*(-2)-(1-2*pL1)*(1-2*pL2))/(pL2*(1-pL1)+pL1*(1-pL2))^2, 0), \\
p2.g2.2 &= c(-1/pL2^2, -(1-2*pL1)^2/(pL2*(1-pL1)+pL1*(1-pL2))^2, -1/(1-pL2)^2), \\
p2.g3.2 &= c(-2/pL2^2, -1/pL2^2-1/(1-pL2)^2, -2/(1-pL2)^2), \\
p1.g4.1 &= c(-1/pL1^2, -(1+2*pL2)^2/(1-pL1-pL2+2*pL1*pL2)^2, -1/(1-pL1)^2), \\
p2.g4.1 &= c(0, ((1-pL1-pL2+2*pL1*pL2)*2-(-1+2*pL2)*(-1+2*pL1))/(1-pL1-pL2+2*pL1*pL2)^2, 0), \\
p1.g4.2 &= c(0, ((1-pL1-pL2+2*pL1*pL2)*2-(-1+2*pL1)*(-1+2*pL2))/(1-pL1-pL2+2*pL1*pL2)^2, 0), \\
p2.g4.2 &= c(-1/(1-pL2)^2, -(-1+2*pL1)^2/(1-pL1-pL2+2*pL1*pL2)^2, -1/pL2^2), \\
p1.g5.1 &= c((((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))*(-2)*(1-rkm)^2 \\
&\quad - (1-rkm)^4*(1-2*pL1)^2)/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2, \\
&\quad (((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2)))*4*(1-rkm)^2 \\
&\quad - (1-rkm)^4*(-2+4*pL1)^2)/((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2)))), \\
&\quad (((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))*(-2)*(1-rkm)^2 \\
&\quad - (1-rkm)^4*(1-2*pL1)^2)/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2), \\
p2.g5.1 &= c(-((1-rkm)^2*(1-2*pL1)*rkm^2*(1-2*pL2))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2, \\
&\quad -(1-rkm)^2*(-2+4*pL1)*rkm^2*(-2+4*pL2))/ \\
&\quad ((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2)))^2, \\
&\quad -((1-rkm)^2*(1-2*pL1)*rkm^2*(1-2*pL2))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2), \\
p1.g5.2 &= c(-(rkm^2*(1-2*pL2)*(1-rkm)^2*(1-2*pL1))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2, \\
&\quad (((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2)))*(-2)*rkm^2 \\
&\quad - rkm^2*(-2+4*pL2)*(1-rkm)^2*(-2+4*pL1))/ \\
&\quad ((1-rkm)^2*(1-2*pL1*(1-pL1)) + rkm^2*(1-2*pL2*(1-pL2)))^2, \\
&\quad -(rkm^2*(1-2*pL2)*(1-rkm)^2*(1-2*pL1))/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2), \\
p2.g5.2 &= c(((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))*(-2)*rkm^2 \\
&\quad - rkm^4*(1-2*pL2)^2)/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2,
\end{aligned}$$


```

      (((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))*4*rkm^2
      - rkm^4*(-2+4*pL2)^2)/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2,

      (((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))*(-2)*rkm^2
      -rkm^4*(1-2*pL2)^2)/((1-rkm)^2*pL1*(1-pL1) + rkm^2*pL2*(1-pL2))^2)
    )
  val<-rep(d2f2,repv)
  if (!is.null(repv2))
    val<-rep(val,repv2)
  val
}

```

B.5 Using the RIM1 functions in batch mode - an example

```

#-----
#to analyse the simulated data, we run R in batch mode.
#example: running R in batch mode, from a UNIX console
#with input file  auto.RIM.r and output file  auto.RIM.log
#-----

R CMD BATCH auto.RIM.r auto.RIM.log &

##### beginning of input file auto.RIM.r #####
#clear workspace.
rm(list=ls())
#all files below should be in the current working directory.
#import data and functions from R data files.
load("b1sim500mid.1qtl.RData")
load("b1.IM.CIM.500mid.1qtl.RData")

#compile functions from text files
source("vuwfunc.r") #utility functions
source("rim.linecross.r") #rim core functions
source("informat.rim.r") #information matrix functions

#-----
# Setup for applying the model to all 100 samples
#-----
test.an.interval<-function(samp,i,nruns,map,mapfun,cross,hypothesis,chosen.model){
  bobject<-paste("b1s",samp, sep="")

```

```

# print(blobject)
val<-rim.linecross(hypothesis=hypothesis,cross=cross,
  data=do.call("$",list(as.name(blobject),"data")),
  regressors=i:(i+1),
  homog.high="AA", heteroz="Aa", homog.low="aa",
  all.markers=do.call("$",list(as.name(blobject),"markers")),
  trait="t1", maxit=nruns,r.curr.next=map,mapfun=mapfun,
  validated=TRUE, chosen.model=chosen.model)
val$data <- c(obj.name=as.name(blobject),dat="data")

val
}

#-----
# To test the 20 intervals, I am going to use 20 computers so it will finish
# running both RIM1 and CIM very quickly for the 100 samples.
# On a Pentium IV 2.6GHz machine with 1024MB RAM, it should take about three hours
# to finish analysing 100 samples of size 500.
#-----

# get host name
system("uname -a")
host <- strsplit(system("echo $HOST",intern=T),"\\\.")[[1]][1]

# host list - these are the computers that I am going to use
host.list <- c("chocolate-days","steamboat","brava","orsinis","the-taj", "circa", "pie-cart",
  "shamiana", "oriental", "quo-vadis", "taputeranga", "stout", "antrim-hse", "mei-kung",
  "aurora", "halswell","wholly-bagels","greta-pt","lone-star","hawkestone" )

host.num<- pmatch(host,host.list)

# make some object names
neut1<-paste("b1c2.LQR.neut",host.num,sep="")
neut0<-paste("b1c2.Q.neut",host.num,sep="")

# We will only be looking at intervals on chromosome two
chrom1.end<-length(b1sim100.map$chrom1[,1])
nsamp<-100
# There are 100 samples.
# Each computer will analyse an interval 100 times by RIM1, and 100 times by CIM...
# and save the output in a file.
print(date())
# -----Fit the RIM1 model for each of the 100 samples-----
# all datasets, b1s1 to b1s100, have the same marker map as b1s1.
assign(neut1,lapply(1:nsamp, test.an.interval, i=chrom1.end+host.num, nruns=200,
  map=b1s1$r.curr.next, mapfun="Haldane", cross="B1", hypothesis="H1",chosen.model="RIM1"))
#neut1 is a list object of length 100, and it stores results for just one interval.

```

```

#Each element in the list neut1 contains the RIM1 output for one sample.
save(list=neut1, file = paste("b1c2.",host.num,".1.Rdata",sep=""))
remove(list=neut1)
#-----Fit the CIM model for each of the 100 samples-----
assign(neut0,lapply(1:nsamp, test.an.interval, i=chrom1.end+host.num, nruns=200,
                    map=b1s1$r.curr.next , mapfun="Haldane", cross="B1", hypothesis="H1",chosen.model="CIM"))
save(list=neut0, file = paste("b1c2.",host.num,".2.Rdata",sep=""))
remove(list=neut0)
print(date())
##### end of file auto.RIM.r #####

```

B.6 Permutation tests with RIM1

```

#-----
#Running some permutations by R in batch mode
#with input file perm.RIM.r and output file perm.RIM.log
#-----

R CMD BATCH perm.RIM.r perm.RIM.log &

##### beginning of input file perm.RIM.r #####
#clear workspace.
rm(list=ls())
#all files below should be in the current working directory.
#import data and functions from R data files.
load("b1sim2000mid.1qtl.RData")
load("b1.IM.CIM.2000mid.1qtl.RData")
#compile functions from text files
source("vwfunc.r") #utility functions
source("rim.linecross.r") #rim core functions
source("informat.rim.r") #information matrix functions

#-----
# The original sample is stored in the object b1s1$data.
# To test the 20 intervals, I am going to use 20 computers.
# Each computer will run 1000 permutations for one testing interval.
#-----
# get host name
system("uname -a")
host <- strsplit(system("echo $HOST",intern=T),"\\\.")[[1]][1]

host.list <- c("cuba","waring-taylor","wakefield","wilton-bush","salamanca",
              "majoribanks","two-rooms","pipitea","rise", "peking-house",

```

```

    "vivian","cafe-laffite","dixon","arizona", "susu",
    "cafe-frenzy","hawkestone","quarter","sfuzzi","la-spaghettata")
hostnum<- pmatch(host,host.list)
print(date())

permuter<-function(h,seed,regressors,data,all.markers,...){
  if (h==1)
    set.seed(seed)
  x<-names(data[,regressors])
  marker.id<- pmatch(x,all.markers)
  if (marker.id[1]==1)
    condLQR<-marker.id[1):(marker.id[1]+2)
  else if (marker.id[2]==length(all.markers))
    condLQR<-(marker.id[2]-2):marker.id[2]
  else condLQR<-(marker.id[1]-1):(marker.id[2]+1)
  conditioning.markers<-all.markers[condLQR]
  MN<- pmatch(x,conditioning.markers)
  KO<-conditioning.markers[-MN]
  #individuals grouped by KO two-locus marker genotype

  g<-apply(cbind(data[,KO]),1,paste,collapse="",sep="")
  indivs<-row.names(data)
  indivs<-split(indivs,g)
  names(indivs)<-NULL
  sorted.rows<-unlist(indivs)
  data<-data[sorted.rows,]

  shuffle<-function(index){
    newseed<-sample(5e7:6e7, 1)
    set.seed(newseed)
    sample(index,replace=FALSE)
  }
  #permute the MN genotypes while keeping everything else fixed
  perm.indivs<-lapply(indivs,shuffle)
  perm.rows<-unlist(perm.indivs)
  perm.data<-data
  perm.data[,x]<-data[perm.rows,x]
  y<-rim.linecross(data=perm.data, regressors=m:(m+1),
    all.markers=b1s1$markers,...)
  y$data<-list(obj.name="b1s1",dat="data")
  y
}

chrom1.end<-length(b1sim100.map$chrom1[,1])
m<-chrom1.end+ hostnum #the right marker
seed<-127+ hostnum #seed for the random number generator

```

```

b1s1.rimperm<-lapply(1:1000,permuter, seed, regressors=m:(m+1),
  data=b1s1$data, all.markers=b1s1$markers, hypothesis="H1", cross="B1",
  homog.high="AA", heteroz="Aa", homog.low="aa", trait=b1s1$traits[1],
  maxit=100, r.curr.next=r.curr.next, mapfun="Haldane",
  chosen.model="RIM1", return.all=T,return.start=F)

print(date())
#make object name
thisperm<-paste("permc2.b1s1.",hostnum,sep="")
assign(thisperm,b1s1.rimperm)
save(list=thisperm,file=paste(thisperm,".Rdata",sep=""))
##### end of file perm.RIM.r #####

```

B.7 Using the RIM1 functions with the Horvat and Medrano mouse data

The following R code shows how the data was converted from QTL cartographer format into an R object suitable for use with the function `rim.linecross()`.

```

#-----
# Horvat and Medrano F2 mouse data
# Data from: Horvat and Medrano, 1995. Genetics 139:1737-1748
# This data was distributed with QTL Cartographer.
# It is the standard QTL Cartographer input format (cross.inp).
# First we use our utility functions to import it into an R list object.
# Then we analyse the data using rim.linecross().
#-----

# import the data
mousec10<-cro.import("D:/vuw4sim/QTLCartWin/example/realdatac.inp")

# import the marker map
wkdir<-"D:/vuw4sim/QTLCartWin/example/"
rmap.call<-paste("Rmap -A -V -W", wkdir, "-i realdatm.inp", "-o mousec10.map -g 3" )
k<-system(rmap.call)
mousec10.c10<-read.table(paste(wkdir,"Chrom.1",sep=""), col.names=c("position.morgans", "chromosome"))
mousec10.map<-list(chrom10=mousec10.c10)
n1<-length(mousec10.map$chrom10[, "position.morgans"])
d.curr.next<-c(mousec10.map$chrom10[2:n1,"position.morgans"]
               -mousec10.map$chrom10[1:(n1-1),"position.morgans"], Inf)
mousec10$r.curr.next<-r.haldane(d.curr.next)
save(list=ls(), file="mousec10.RData")
# finished importing and configuring the data.

```

The following R code shows how the Horvat and Medrano mouse data was analysed using the function `rim.linecross()`.

```
#-----
# load the Horvat and Medrano F2 mouse data and fit the RIM1 model
#-----

load("mousec10.RData")

#the next line runs RIM1 on all eight intervals
mousec10.rim1<-lapply(1:8, function(h,...){
  y<-rim.linecross(regressors=h:(h+1),...)
  y$data<-list(obj.name="mousec10", dat="data"); y}, hypothesis="H1", cross="F2",
  data=mousec10$data, homog.high="AA", heteroz="Aa", homog.low="aa", all.markers=mousec10$markers,
  trait=mousec10$traits[1],maxit=100, r.curr.next=mousec10$r.curr.next, mapfun="Haldane",
  chosen.model="RIM1", return.all=T, return.start=F)

#-----
# Finished fitting the model. Now summarise the results.
#-----

# this is a small dataset, so print the raw output for all intervals.
print(mousec10.rim1)

#To summarise the results, first determine whether a QTL was detected in each interval
testit.f2<-function(ml,siglevel,bQonly=FALSE){
  if (bQonly){
    pv.effect.a<-ml$mle$model.params$effects["Q.aAA","P>|z0|"]
    pv.effect.d<-ml$mle$model.params$effects["Q.dAA","P>|z0|"]
    detected<-FALSE
    if ((pv.effect.a<siglevel) ||(pv.effect.d<siglevel))
      detected<-TRUE
  }
  else{
    pv.effect.a<-ml$mle$model.params$effects["Q.aAA","P>|z0|"]
    pv.effect.d<-ml$mle$model.params$effects["Q.dAA","P>|z0|"]
    pv.interior<-ml$mle$model.params$probs["pQ2",c("P>z0","P<z1")]
    detected<-FALSE
    if ((max(c(pv.interior,pv.effect.a))<siglevel)
        ||(max(c(pv.interior,pv.effect.d))<siglevel))
      detected<-TRUE
  }
  detected
}

#test QTL effect and position at the 5% significance level
power.rim1.05<-unlist(lapply(mousec10.rim1,testit.f2,0.05,F))
```

```

print(power.rim1.05)
#-----
#for ease of inspection, collect the results into a list/matrix
#-----
getf2mle<-function(h){
  pv.effect.a<-h$mle$model.params$effects["Q.aAA","P>|z0|"]
  pv.effect.d<-h$mle$model.params$effects["Q.dAA","P>|z0|"]
  pv.interior<-max(h$mle$model.params$probs["pQ2",c("P>z0","P<z1")])
  sd.aQ<-h$mle$model.params$effects["Q.aAA",2]
  sd.dQ<-h$mle$model.params$effects["Q.dAA",2]
  sd.pQ2<-h$mle$model.params$probs["pQ2",2]
  val<-c(h$mle$model.params$effects["Q.aAA",1], h$mle$model.params$effects["Q.dAA",1],
        h$mle$model.params$probs["pQ2",1], pv.effect.a,pv.effect.d, pv.interior,
        h$mle$model.params$variance, h$mle$recomb["rMQ"], h$mle$recomb["rMN"],
        h$mle$loglike, sd.aQ, sd.dQ, sd.pQ2)
  val
}

RIM1.m10.all<-t(sapply(mousec10.rim1,getf2mle))
dimnames(RIM1.m10.all)[[2]]<- c("aQ","dQ","pQ2","pv.aQ","pv.dQ","pv.interior","sigma2",
                                "rMQ","rMN","loglike", "sd.aQ","sd.dQ","sd.pQ2")
RIM1.m10.all<-as.data.frame(RIM1.m10.all)
RIM1.m10.all$sig.effect<-""
RIM1.m10.all$sig.effect[(RIM1.m10.all$pv.aQ<0.05)|(RIM1.m10.all$pv.dQ<0.05)]<-"*"
RIM1.m10.all$sig.effect[(RIM1.m10.all$pv.aQ<0.01)|(RIM1.m10.all$pv.dQ<0.01)]<- "***"
RIM1.m10.all$sig.effect[(RIM1.m10.all$pv.aQ<0.001)|(RIM1.m10.all$pv.dQ<0.001)]<- "****"

RIM1.m10.all$sig.interior<-""
RIM1.m10.all$sig.interior[RIM1.m10.all$pv.interior<0.05]<- "*"
RIM1.m10.all$sig.interior[RIM1.m10.all$pv.interior<0.01]<- "***"
RIM1.m10.all$sig.interior[RIM1.m10.all$pv.interior<0.001]<- "****"

nmarkers<-length(mousec10$markers)
bmnames<-paste(paste(mousec10$markers[1:(nmarkers-1)],sep=""),"-",
               paste(mousec10$markers[2:nmarkers],sep=""),sep="")
row.names(RIM1.m10.all)<-bmnames

x<-RIM1.m10.all[, c("aQ","dQ","pQ2","rMQ","rMN", "sd.aQ","sd.dQ","sd.pQ2","pv.aQ","pv.dQ",
                  "sig.effect","pv.interior", "sig.interior")]
x<-as.data.frame(x)
dMQ<-d.haldane(x$rMQ)
dMN<-d.haldane(mousec10$r.curr.next[-nmarkers])

#calculate confidence intervals for pQ2
ci.calc<-function(a,pQ2,rmn,sd.pQ2){

```

```

        pp<-1-a/2
        sd.pQ2[sd.pQ2==0]<-NA
        pQ2.l<- pQ2-qnorm(pp)*sd.pQ2
        pQ2.l[pQ2.l<0]<-0
        pQ2.u<- pQ2+qnorm(pp)*sd.pQ2
        pQ2.u[pQ2.u>1]<-1
        ci.pQ2<-cbind(lower=pQ2.l,upper=pQ2.u,rmn=rmn)
        ci.pQ2
    }

ci.pQ2<-ci.calc(0.01,x[, "pQ2"],x[, "rMN"],x[, "sd.pQ2"])

pQ2torMQ<-function(pQ2,rmn){
    pQ1<-(0.5+sqrt(1-2*rmn+rmn^2*(1-2*pQ2)^2)/(2*(1-rmn)))
    if (any(pQ1>1)) print("yes")
    rMQ<-((1-rmn)*(1-pQ1)+rmn*(1-pQ2))
}

#as pQ2 increases rMQ decreases
#calculate confidence intervals for rMQ
ci.rMQ<-cbind(lower=pQ2torMQ(ci.pQ2[, "upper"],ci.pQ2[, "rmn"]),
    upper=pQ2torMQ(ci.pQ2[, "lower"],ci.pQ2[, "rmn"]))
#distance in centi Morgans
ci.dMQ<-d.haldane(ci.rMQ)

#collate all results into a tabular object
x1<-data.frame(aQ=round(x$aQ,2),sd.aQ=round(x$sd.aQ,2),
dQ=round(x$dQ,2),sd.dQ=round(x$sd.dQ,2),
    pv.aQ=round(x$pv.aQ,4),pv.dQ=round(x$pv.dQ,4),
    sig.bQ=x$sig.effect,dMN=round(dMN*100,2),
    dMQ=round(dMQ*100,1),
    pv.pQ2=round(x$pv.interior,4),sig.pQ2=x$sig.interior,
    low.dMQ=round(ci.dMQ[,1]*100,1),
    up.dMQ=round(ci.dMQ[,2]*100,1))
row.names(x1)<-bmnames
print(x1)

#check whether information matrix was non singular in all intervals
check.imat<-sapply(mousec10.rim1,function(h){h$mle$infmtat.is.singular})
#how many times was a singular information matrix encountered
length(check.imat[check.imat==TRUE])
#Answer was zero times for this dataset. [finished]
save(list=ls(), file="results.mousec10.RData")

```


References

- Aitchison, J. and Silvey, S. D. (1960). Maximum-likelihood estimation procedures and associated tests of significance. *J. Roy. Statist. Soc. ser. B* **22**: 154–171.
- Akaike, H. (1974). A new look at statistical model identification. *IEEE Transactions on Automatic Control* **AU-19**: 716–722.
- Anderson, J. A. (1979). Multivariate logistic compounds. *Biometrika* **66**: 17–26.
- Andrews, D. W. K. (1999). Estimation when a parameter is on a boundary. *Econometrica* **67**(6): 1341–1383.
- Arondel, V., Lemieux, B., Hwang, I., Gibson, S., Goodman, H. M. and Somerville, C. R. (1992). Map-based cloning of a gene controlling omega-3 fatty acid desaturation in arabidopsis. *Science* **258**: 1353–1355.
- Ball, R. D. (2001). Bayesian methods for quantitative trait loci mapping based on model selection: approximate analysis using the Bayesian Information Criterion. *Genetics* **159**: 1351–1364.
- Basford, K. E., Grenway, D. R., McLachlan, G. J. and Peel, D. (1997). Standard errors of fitted component means of normal mixtures. *Computational Statistics* **12**: 1–17.

- Basford, K. E. and McLachlan, G. J. (1985). Likelihood estimation with Normal mixture models. *Applied Statistics* **34**(3): 282–289.
- Basten, C. J., Weir, B. S. and Zeng, Z.-B. (1994). *Zmap-a QTL cartographer*. In: *Proceedings of the 5th World Congress on Genetics Applied to Livestock Production: Computing Strategies and Software* edited by C. Smith, J. S. Gavora, B. Benkel, J. Chesnais, W. Fairfull, J. P. Gibson, B. W. Kennedy and E. B. Burnside., The Organizing Committee, 5th World Congress on Genetics Applied to Livestock Production, Guelph, Ontario, Canada, volume 22, 65–66.
- Basten, C. J., Weir, B. S. and Zeng, Z.-B. (2001). *QTL Cartographer: A reference manual and tutorial for QTL mapping*. Department of Statistics, North Carolina State University, Raleigh, NC. URL <http://statgen.ncsu.edu/qtlcart/>. Version 1.15.
- Behboodian, J. (1972a). Information matrix for a mixture of two Normal distributions. *Journal of Statistical Computation and Simulation* **1**(4): 295–315.
- Behboodian, J. (1972b). On the distribution of a symmetric statistic from a mixed population. *Technometrics* **14**(4): 919–923.
- Behboodian, J. (1973). Information matrix for a mixture of two exponential distributions. *Journal of Statistical Computation and Simulation* **2**(1): 1–17.
- Belknap, J. K. (1998). Effect of within-strain sample size on qtl detection and mapping using recombinant inbred mouse strains. *Behavior Genetics* **28**(1): 29–38.
- Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc. B* **57**(1): 289–300.

- Beran, R. and Ducharme, G. R. (1991). *Asymptotic theory for bootstrap methods in statistics*. Les Publications CRM, Université de Montréal, Canada.
- Bickel, P. J. and Freedman, D. A. (1981). Some asymptotics on the bootstrap. *Ann. Statist.* **9**: 1196–1217.
- Bohn, M., Groh, S., Khairallah, M. M., Hoisington, D. A., Utz, H. F. and Melchinger, A. E. (2001). Re-evaluation of the prospects of marker-assisted selection for improving insect resistance against diatraea spp. in tropical maize by cross validation and independent validation. *Theor. Appl. Genet.* **103**: 1059–1067.
- Böhning, D., Schlattman, P. and Lindsay, B. G. (1992). Computer assisted analysis of mixtures (CAMAN): Statistical algorithms. *Biometrics* **48**: 283–304.
- Breusch, T. S. (1986). Hypothesis testing in unidentified models. *Review of Economic Studies* **53**(4): 635–651.
- Bridges, W. C. and Knapp, S. J. (1990). Mapping quantitative trait loci using molecular marker linkage maps. *Theor. Appl. Genet.* **79**: 583–592.
- Cadogan, C. C. (1987). *Elements of Mathematical Structures*. Learning Resource Center, University of the West Indies, Barbados.
- Cannings, C., Thompson, E. A. and Skolnik, M. H. (1978). Probability functions on complex pedigrees. *Adv. Appl. Probab.* **10**: 26–61.
- Carlin, B. and Lewis, T. (1996). *Bayes and Empirical Bayes Methods for Data Analysis*. Chapman & Hall/CRC, USA.
- Casella, G. and George, E. I. (1992). Explaining the Gibbs Sampler. *The American Statistician* **46**: 167–174.

- Chang, M. N., Wu, R., Wu, S. S. and Casella, G. (2003). Score statistics for mapping quantitative trait loci. Technical report, University of Florida.
- Chen, H. and Chen, J. (1998a). The likelihood ratio test for homogeneity in finite mixture models. Technical Report STAT 98-08, University of Waterloo.
- Chen, H. and Chen, J. (1998b). The likelihood reatio test for homogeneity in normal mixtures with presence of a structural parameter. Technical Report STAT 98-09, University of Waterloo.
- Chen, H., Chen, J. and Kalbfleish, J. D. (2001). A modified likelihood ratio test for homogeneity in finte mixture models. *J. Roy. Statist. Soc. ser. B* **63**: 19–29.
- Chung, J., Babka, H. L., Graef, G. L., Staswick, P. E., Lee, D. J., Cregan, P. B., Shoemaker, R. C. and Specht, J. E. (2003). The seed protein, oil and yield QTL on soybean linkage group I. *Crop Science* **43**: 1053–1067.
- Churchill, G. A. and Doerge, R. W. (1994). Empirical threshold values for quantitative trait locus mapping. *Genetics* **138**: 963–971.
- Cockerham, C. C. (1954). An extension of the concept of partitioning hereditary variance for analysis of covariances among relatives when epistasis is present. *Genetics* **39**: 859–882.
- Corander, J. and Sillanpää, M. J. (2002). A unified approach to joint modeling of multiple quantitative and qualitative traits in gene mapping. *Journal of Theoretical Biology* **218**(4): 435–446.
- Cox, D. R. and Hinkley, D. V. (1974). *Theoretical Statistics*. Chapman and Hall, London.

- Davarsi, A. and Soller, M. (1994). Optimum spacing of genetic markers for determining linkage between marker loci and quantitative trait loci. *Theor. Appl. Genet.* **89**: 351–357.
- Davarsi, A. and Soller, M. (1995). Advanced Intercrossed Lines, an experimental population for fine genetic mapping. *Genetics* **141**: 1199–1207.
- Davies, R. B. (1977). Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika* **64**: 247–254.
- Davies, R. B. (1987). Hypothesis testing when a nuisance parameter is present only under the alternative. *Biometrika* **74**(1): 33–43.
- Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. Roy. Statist. Soc. ser. B* **39**: 1–22.
- Diebolt, J. and Robert, C. (1994). Estimation of finite mixture distributions through Bayesian sampling. *J. Roy. Statist. Soc. ser. B* **56**: 363–375.
- Doerge, R. W., Zeng, Z.-B. and Weir, B. S. (1997). Statistical issues in the search for genes affecting quantitative traits in experimental populations. *Statistical Science* **12**(3): 195–219.
- Draper, N. R. and Smith, H. (1998). *Applied Regression Analysis*. John Wiley and Sons Ltd., USA, third edition.
- Edwards, J. H. (1987). The locus ordering problem. *Ann. Hum. Genet.* **51**: 251–258.
- Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Ann. Statist.* **7**: 1–26.
- Elston, R. C. and Stewart, J. (1971). A general model for the genetic analysis of pedigree data. *Hum. Hered.* **21**: 523–542.

- Everitt, B. S. and Hand, D. J. (1981). *Finite mixture distributions*. Chapman and Hall, USA.
- Falconer, D. S. and Mackay, T. F. C. (1996). *Introduction to Quantitative Genetics*. Longman, UK, fourth edition.
- Fisher, R. A. (1918). The correlation between relatives on the supposition of mendelian inheritance. *Trans. Royal. Soc. Edinburgh* **52**: 399–433.
- Frankel, W. N. (1995). Taking stock of complex trait genetics in mice. *Trends in Genetics* **11**(12): 471–477.
- Gelderman, H. (1975). Investigations on inheritance of quantitative characters by gene markers. *Theor. Appl. Genet.* **46**: 319–330.
- Ghosh, J. M. and Sen, P. K. (1985). *On the asymptotic performance of the log likelihood ratio statistic for the mixture model and related results*. In: *Proc. Berkeley Conference in Honour of Jersey Neyman and Jack Keifer* edited by L. M. Hartigan and R. A. Olshen, Wadsworth, Monterey, volume 2, 789–806.
- Goffinet, B. and Gerber, S. (2000). Quantitative trait loci: A meta-analysis. *Genetics* **155**: 463–473.
- Guo, S. W. and Thompson, E. A. (1992). A Monte Carlo method for combined segregation and linkage analysis. *Am. J. Hum. Genet.* **51**: 1111–1126.
- Hackett, C. A. and Weller, J. I. (1995). Genetic mapping of quantitative trait loci for traits with ordinal distributions. *Biometrics* **51**(4): 1254–1263.
- Haldane, J. B. S. (1919). The combination of linkage values and the calculation of distances between the loci of linked factors. *Journal of Genetics* **8**: 299–309.

- Haley, C. S. and Knott, S. A. (1992). A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. *Heredity* **69**: 315–324.
- Hartl, D. L. and Clark, A. G. (1997). *The Principles of Population Genetics*. Sinauer, Canada.
- Hayes, P. M., Liu, B. H., Knapp, S. J., Chen, F., Jones, B., Blake, T., Franckowiak, J., Rasmusson, D., Sorrells, M., Ullrich, S. E., Wesenberg, D. and Kleinjans, A. (1993). Quantitative trait locus effects and environmental interaction in a sample of North American barley germplasm. *Theor. Appl. Genet.* **87**: 392–401.
- Hill, B. M. (1963). Information for estimating the proportions in mixtures of exponential and normal distributions. *Journal of the American Statistical Association* **58**(304): 918–932.
- Hochberg, Y. and Tamhane, A. C. (1987). *Multiple comparison procedures*. John Wiley and Sons, USA.
- Hoeschele, I., Uimari, P., Grignola, F. E., Zhang, Q. and Gage, K. M. (1997). Advances in statistical methods to map quantitative trait loci in outbred populations. *Genetics* **147**: 1445–1457.
- Horvat, S. and Medrano, J. F. (1995). Interval mapping of high growth (hg), a major locus that increases weight gain in mice. *Genetics* **136**: 1737–1748.
- Jaiswal, P., Ni, J., Yap, I., Ware, D., Spooner, W., Youens-Clark, K., Ren, L., Liang, C., Hurwitz, B., Zhao, W., Ratnapu, K., Faga, B., Canaran, P., Fogleman, M., Heberd, C., Avraham, S., Schmidt, S., Casstevens, T., S, E., Buckler, Stein, L. and McCouch, S. (2006a). Gramene: a genomics and genetics resource for rice. *Rice Genetics Newsletter* **22**(1): 9–16.

- Jaiswal, P., Ni, J., Yap, I., Ware, D., Spooner, W., Youens-Clark, K., Ren, L., Liang, C., Zhao, W., Ratnapu, K., Faga, B., Canaran, P., Fogleman, M., Hebbard, C., Avraham, S., Schmidt, S., Casstevens, T. M., Buckler, E. S., Stein, L. and McCouch, S. (2006b). Gramene: a bird's eye view of cereal genomes. *Nucleic Acids Research* **34**(1): D717–D723. Database issue.
- Jander, G., Norris, S. R., Rounsley, S. D., Bush, D. F., Levin, I. M. and Last, R. L. (2002). Arabidopsis map-based cloning in the post-genome era. *Plant Physiology* **129**: 440450.
- Jansen, R. C. and Stam, P. (1994). High resolution of quatitative traits into multiple loci via interval mapping. *Genetics* **136**: 1447–1455.
- Jiang, C. and Zeng, Z.-B. (1995). Multiple trait analysis of genetic mapping for quantitative trait loci. *Genetics* **140**: 1111–1127.
- Jiang, C. and Zeng, Z.-B. (1997). Mapping qunatitative trait loci with dominant and missing markers in varaious crosses from two inbred lines. *Genetica* **101**: 47–58.
- Kao, C. H. and Zeng, Z.-B. (1997). General formulas for obtaining the MLEs and the asymtotic variance-covariance matrix in mapping quantitative trait loci when using the EM alogrithm. *Biometrics* **53**: 653–665.
- Kao, C. H., Zeng, Z.-B. and Teasdale, R. D. (1999). Multiple interval mapping for quantitative trait loci. *Genetics* **152**: 1203–1216.
- Kearse, M. J. and Hyne, V. (1994). QTL analysis: a simple "marker regression" approach. *Theor. Appl. Genet.* **90**: 698–702.
- Kempthorne, O. (1954). The correlation between relatives in a random mating population. *Proc. Royal Soc. Lond. B* **143**: 103–113.

- Knott, S. A. and Haley, C. S. (1992). Aspects of maximum likelihood methods for the mapping of quantitative trait loci in line crosses. *Genetical Research* **60**: 139–161.
- Kosambi, D. D. (1944). The estimation of map distances from recombination values. *Ann. Eugen.* **12**: 172–175.
- Kruglyak, L. and Lander, E. S. (1995). A nonparametric approach for mapping quantitative trait loci. *Genetics* **139**: 1421–1428.
- Kuhl, K. P. and Giardina, C. R. (1982). Elliptic Fourier features of a closed contour. *Computer Graphics and Image Processing* **18**: 236–258.
- Lander, E. S. and Botstein, D. (1989). Mapping Mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics* **121**: 185–199.
- Lehmann, E. L. and Casella, G. (1998). *Theory of point estimation*. Springer-Verlag, USA, second edition.
- Lesperance, M. L. and Lindsay, B. G. (2001). Statistical computing in mixtures. Technical Report #01-07-13, The Pennsylvania State University.
- Lincoln, S., Daley, M. and Lander, E. (1992). Mapping genes controlling quantitative traits with MAPMAKER/QTL 1.1. Technical Report 2nd edition, Whitehead Institute.
- Lindsay, B. G. (1995). *Mixture Models: Theory Geometry and Applications*. Institute of Mathematical Statistics, American Statistical Association, USA.
- Liu, B. H. (1997). *Statistical genomics: linkage mapping and QTL analysis*. CRC Press, USA.

- Liu, J., Mercer, J. M., Stam, L. F., Gibson, G. C., Zeng, Z.-B. and Laurie, C. C. (1996). Genetic analysis of a morphological shape difference in the male genitalia of *drosophila simulans* and *d. mauritiana*. *Genetics* **142**: 1129–1145.
- Louis, T. A. (1982). Finding the observed information matrix when using the em algorithm. *J. R. Statist. Soc. B* **44**(2): 226–233.
- Lynch, M. and Walsh, B. (1997). *Genetics and analysis of quantitative traits*. Sinauer, Canada.
- Manly, B. F. (1997). *Randomization, Bootstrap and Monte Carlo Methods in Biology (second edition)*. Chapman and Hall, USA.
- Mardia, K. V., Kent, J. T. and Bibby, J. M. (1979). *Multivariate Analysis. Probability and Mathematical Statistics*. Academic Press Ltd., London, UK.
- Marinez, O. and Curnow, R. N. (1992). Estimating the locations and the sizes of the effects of quantitative trait loci using flanking markers. *Theor. Appl. Genet.* **85**: 480–488.
- McLachlan, G. J. and Basford, K. E. (1987). *Mixture models: inference and applications to clustering*. Marcel Dekker Inc., USA.
- McLachlan, G. J. and Krishnan, T. (1996). *The EM Algorithm and Extensions*. John Wiley and Sons Inc., USA.
- Milhaljevic, R., Utz, H. F. and Melchinger, A. E. (2004). Congruency of quantitative trait loci detected for agronomic traits in testcrosses of five populations of european maize. *Crop Science* **44**: 114–124.
- Montgomery, D. C. (1996). *Design and analysis of experiments*. John Wiley and Sons Ltd., USA, fourth edition.

- Morgante, M. and Salamini, F. (2003). From plant genomics to breeding practice. *Current Opinion in Biotechnology* **14**: 214–219.
- O'Connell, J. R. and Weeks, D. E. (1995). The vitesse algorithm for rapid exact multilocus linkage analysis via genotype set-recording and fuzzy inheritance. *Nat. Genet.* **11**: 402–408.
- Ott, J. (1991). *Analysis of human genetic linkage, Revised Edition*. Johns Hopkins, London.
- Otto, S. P. and Jones, C. D. (2000). Detecting the undetected: Estimating the total number of loci underlying a trait. *Genetics* **156**: 2093–2107.
- Paterson, A. H., Lander, E. S., Hewitt, J. D., Peterson, S., Lincoln, S. E. and Tanksley, S. D. (1988). Resolution of quantitative traits into mendelian factors by using a complete RFLP linkage map. *Nature* **335**: 721–726.
- Piepho, H. P. (2000). A mixed-model approach to mapping quantitative trait loci in barley on the basis of multiple environment data. *Genetics* **156**: 2043–2050.
- Prescott, P., Dean, A. M., Draper, N. R. and Lewis, S. M. (2002). Mixture models: Ill conditioning and quadratic model specification. *Technometrics* **44**(3): 260–268.
- Price, A. H., Young, E. M. and Tomos, A. D. (1997). Quantitative trait loci associated with stomatal conductance, leaf rolling and heading date mapped in upland rice (*Oryza sativa*). *New Phytologist* **137**: 83–91.
- R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org>. ISBN 3-900051-07-0.

- Rao, C. R. and Toutenburg, H. (1995). *Linear models: least squares and alternatives*. Springer-Verlag, USA.
- Redner, R. A. (1981). Note on the consistency of the maximum-likelihood estimate for non-identifiable distributions. *Ann. Statist.* **9**: 225–228.
- Redner, R. A. and Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Rev.* **26**: 195–239.
- Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with an unknown number of components. *J. Roy. Statist. Soc. ser. B* **59**(4): 731–792.
- Robert, C. P. (1996). *Mixtures of distributions: inference and estimation*. In: *Markov Chain Monte Carlo in Practice* edited by R. W. Gilks, S. Richardson and D. J. Spiegelhalter, Chapman and Hall, London, 441–463.
- Robertson, T., Wright, F. T. and Dykstra, R. L. (1988). *Order restricted statistical inference*. John Wiley and Sons, UK.
- Rodolphe, F. and Lefort, M. (1993). A multiple-marker approach for detecting chromosomal segments displaying QTL activity. *Genetics* **134**: 1227–1288.
- Rotnitzky, A., Cox, D. R., Bottai, M. and Robins, J. (2000). Likelihood based inference with a singular information matrix. *Bernoulli* **6**: 243–284.
- Satagopan, J. M. and Yandell, B. S. (1996). Estimating the number of quantitative trait loci via bayesian model determination. URL <ftp://ftp.stat.wisc.edu/pub/yandell/revjump.html>. Special contributed paper, session on genetic analysis of quantitative traits and complex diseases. Biometric Section, Joint Statistical Meetings, Chicago.

- Satagopan, J. M., Yandell, B. S., Newton, M. A. and Osborn, T. C. (1996). A Bayesian approach to detect quatitative trait loci using Markov Chain Monte Carlo. *Genetics* **144**: 805–816.
- Schoenberg, R. (2001). Constrained optimization (white paper). Aptech Systems, Inc., Maple Valley, WA, USA.
- Seidel, W., Mosler, K. and Alker, M. (2000). A cautionary note on likelihood ratio tests in mixture models. *Annals of the Institute of Statistical Mathematics* **52**: 481–487.
- Self, S. and Liang, K.-Y. (1987). Asymptotic properties of maximum likelihood estimators in non-standard conditons. *Journal of the American Statistical Association*, **82**(398): 605–610.
- Silvey, S. D. (1959). The lagrangian multiplier test. *Annals of Mathematical Statistics* **30**: 389–407.
- Smith, A. F. M. and Roberts, G. O. (1993). Bayesian computation via Gibbs sampler and related Markov Chain Monte Carlo methods (with discussion). *J. Roy. Statist. Soc. Ser. B* **55**: 3–23.
- Soller, M., Brody, T. and Genizi, A. (1976). On the power of experimental design for the detection of linkage between marker loci and quantitative loci in crosses between inbred lines. *Theor. Appl. Genet.* **47**: 35–39.
- Soller, M. and Medjugorac, I. (1999). A successful marriage: Making the transition from QTL mapping to marker-assisted selection. In Dekkers, J. C. M., Lamont, S. J. and Rothschild, M. F., editors, *From Jay Lush to Genomics: Visions For Animal Breeding And Genetics*. May 16-18, 1999, Iowa State University, Ames, Iowa USA. <http://www.agbiotechnet.com/proceedings/jaylush.asp>.

- Spelman, R. J., Huisman, A. E., Singireddy, S. R., Coppieters, W., Arranz, J., Georges, M. and Garrick, D. J. (1999). Short communication: quantitative trait loci analysis on 17 nonproduction traits in the new zealand dairy population. *J. Dairy Sci.* **82**: 2514–2516.
- Stuber, C. W., Edwards, M. D. and Wendel, J. F. (1987). Molecular -marker-facilitated investigations of quantitative-trait loci in maize II. Factors influencing yield and its component traits. *Crop Science* **27**: 639–648.
- Swanepoel, J. W. H. (1986). A note on proving that the (modified) bootstrap works. *Comm. Statist. Theory Methods* **15**: 3193–3203.
- Tanksley, S. D., Ganai, M. W. and Martin, G. B. (1995). Chromosome landing: a paradigm for map-based cloning in plants with large genomes. *Trends in Genetics* **11**: 63–68.
- Thoday, J. M. (1961). Location of polygenes. *Nature* **191**: 368–370.
- Titterton, D. M. (1981). In discussion of M. Aitkin, D. Anderson and J. Hinde. *J. R. Statist. Soc. A* **144**: 459.
- Titterton, D. M., Smith, A. F. M. and Makov, U. E. (1985). *Statistical analysis of finite mixture distributions*. John Wiley and Sons, UK.
- Venables, W. N. and Ripley, B. D. (1997). *Modern applied statistics with S-PLUS*. Springer-Verlag, USA, second edition.
- Visscher, P. M., Haley, C. S. and Knott, S. A. (1996a). Mapping QTLs for binary traits in backcross and F₂ populations. *Genetical Research* **68**: 55–63.
- Visscher, P. M., Thompson, R. and Haley, C. S. (1996b). Confidence intervals in QTL mapping by bootstrapping. *Genetics* **143**: 1013–1020.

- Visscher, P. M., Thompson, R. and Haley, C. S. (2000). Bias and sampling error of the estimated proportion of genotypic variance explained by quantitative trait loci determined from experimental data in maize using cross validation and validation with independent samples. *Genetics* **154**: 1839–1849.
- Vladutu, C., McLaughlin, J. and Phillips, R. L. (1999). Fine mapping and characterization of linked quantitative trait loci involved in the transition of the maize apical meristem from vegetative to generative structures. *Genetics* **153**: 993–1007.
- Vu, H. T. V. and Zhou, S. (1997). Generalization of likelihood ratio tests under nonstandard conditions. *The Annals of Statistics* **25**(2): 897–916.
- Warden, C. H., Fisler, J. S., Pace, M. J., Svenson, K. L. and Lusis, A. J. (1993). Coincidence of genetic loci for plasma cholesterol levels and obesity in a multifactorial mouse model. *Journal of Clinical Investigation* **92**(2): 773–779.
- Ware, D., Jaiswal, P., Ni, J. J., Pan, X., Chang, K., Clark, K., Teytelman, L., Schmidt, S., Zhao, W., Cartinhour, S., McCouch, S. R. and Stein, L. (2002). Gramene: a resource for comparative grass genomics. *Nucleic Acids Research* **30**(1): 103–105.
- Weller, J. I. (1986). Maximum likelihood techniques for the mapping and analysis of quantitative trait loci with the aid of genetic markers. *Biometrics* **42**: 627–640.
- Whittaker, J. C., Thompson, R. and Visscher, P. (1996). On the mapping of QTL by regression of phenotype on marker type. *Heredity* **77**: 23–32.
- Williams, R. W., Broman, K. W., Cheverud, J. M., Churchill, G. A., Hitzemann, R. W., Hunter, R. W., Mountz, J., Pomp, D., Reeves, R. H., Schalkwyk, L. C. and Threadgill, D. W. (2002). A collaborative cross for high-precision complex

- trait analysis. 1st Workshop Report of the Complex Trait Consortium: September 2002. http://www.complextait.org/pdf_files/CTC_Workshop_v31.pdf.
- Xu, S. (2003). Theoretical basis of the Beavis effect. *Genetics* **165**: 2259–2268.
- Xu, X., Fang, Z., Wang, B., Chen, C., Guang, W., Jin, Y., Yang, J., Lewitzky, S., Aelony, A., Parker, A., Meyer, J., Weiss, S. T. and Xu, X. (2001). A genomewide search for quantitative-trait loci underlying asthma. *Am. J. Hum. Genet.* **69**: 1271–1277.
- Yi, N., George, V. and Allison, D. B. (2003). Stochastic search variable selection for identifying multiple quantitative trait loci. *Genetics* **164**: 1129–1138.
- Zeng, Z.-B. (1993). Theoretical basis for the separation of multiple linked gene effects on mapping quantitative trait loci. *Proc. Natl. Acad. Sci. USA* **90**: 10972–10976.
- Zeng, Z.-B. (1994). Precision mapping of quantitative trait loci. *Genetics* **136**: 1457–1466.
- Zeng, Z.-B., Liu, J., Stam, L. F., Kao, C.-H., Mercer, J. M. and Laurie, C. C. (2000). Genetic architecture of a morphological shape difference between two drosophila species. *Genetics* **154**: 299–310.
- Zeng, Z.-B., Wang, T. and Zou, W. (2005). Modelling quantitative trait loci and interpretation of models. *Genetics* **169**: 1711–1725.
- Zou, F. (2001). *Efficient and Robust Statistical Methodologies for Quantitative Trait Loci Analysis*. Ph.D. thesis, University of Wisconsin - Madison.