EFFICIENT COMPUTATIONAL GENETICS METHODS FOR MULTIPARENT CROSSES

Zhaojun Zhang

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science

Chapel Hill
2014

Approved by:

Wei Wang

William Valdar

Leonard McMillan

Fernando Pardo Manuel de Villena

Vladimir Jojic

ABSTRACT

ZHAOJUN ZHANG: Efficient Computational Genetics Methods for Multiparent
Crosses
(Under the direction of Wei Wang and William Valdar)

Multiparent crosses are genetic populations bred in a controlled manner from a finite number of known founders. They represent experimental resources that are of potentially great value for understanding the genetic basis of complex diseases. An important new experimental technology that can be applied to multiparent crosses, namely high-throughput sequencing, generates an immense amount of data and provides unprecedented opportunities to study genetics at a ultra high resolution. However, to take advantage of such massive data, several computational genetics problems have to be resolved. These include RNA-Seq assembly and quantification, QTL mapping, and haplotype effect estimation. In order to tackle these problems, which are highly connected to each other, I propose a series of methods: GeneScissors is a novel method to detect errors caused by multiple alignments in the RNA-Seq; RNA-Skim can rapidly quantify RNA-Seq data while still provide reliable results; HTreeQA is designed as a phylogeny based QTL mapping method for genotypes with heterozygou sites; and Diploffect estimates founder effects with statistically valid interval estimates in multiparent crosses. These methods are extensively studied on both simulated and real data. These studies demonstrate that the proposed methods can make data analysis of multiparent crosses more effective and efficient and produce results are more accurate and trustworthy than a number of existing alternative methods.

To

Liping Zhang and Aimin Zhang

Jingjing Sun

for your love and support

ACKNOWLEDGEMENTS

I am indebted to my advisor, Professor Wei Wang, for her guidance and support over the last six years. She led me into the research world of computational genetics, taught me both research skills and computer science skills, encouraged me during tough times in the Ph.D. pursuit, ensured that I worked on my research productively, collaboratively and independently, and worked with me by contributing her time, ideas, and funding. I am also thankful for the excellent example she has provided as a successful computer scientist and professor.

I am also indebted to my second advisor, Professor William Valdar over the last four and a half years. Will guided me how to be a humble researcher, how to develop rigorous Bayesian models, and how to be critical and skeptical of my own research in a constructive way. He also supported me by contributing his time, ideas, and funding. I will miss the meetings we had in the past years, because I always learned something new from the meetings.

I would like to thank Professor Leonard McMillan and Fernando Pardo Manuel de Villena for not only serving on my thesis committee, but also, for introducing me and educating me into the world of computational genetics and for collaborating with me on several projects.

I would like to thank Professor Vladimir Jojic for serving on my thesis committee, carefully examining my research and writing, and offering insightful feedbacks from general research discussions to detailed writing advice.

My thanks to all computational genetics group members and Valdar Lab members, in particular to Feng Pan, Xiang Zhang, Qi Zhang, Ning Jin, Eric Liu Yi, Shunping Huang, Weibo Wang, Wei Cheng, Summer Goodson, Yuying Xie, Alan Lenarcic, Dan

Oreper, Robert Corty, Jeremy Sabourin, Greg Keele for all the valuable discussions.

I would like to thank my parents for their love and encouragement. It was a hard time for them to let their only child be aboard for six years, but they showed me nothing but their truly and unconditional support. I would like to thank Jingjing's parents Jinghua Song and Pingyu Sun, who educated me and helped me to be a better man. Lastly, I want to thank my loving, supportive, and patient fiancée Jingjing Sun, who always believe that I can reach the finishing line, and thank her for being an important part of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In modern genetics, the main doctrine is that DNA encodes the genetic information that contributes to the variations of inherited characteristics. The ultimate goal of genetics is to decipher how the information in the genome influences traits and/or diseases. The pedigree of Humans is deeply rooted, extremely mixed, and composed by thousands of generations, requiring arduous efforts to understand the relationship between genetic information and observable traits. Hence, model organisms, such as mouse, abrabidopsis, and maize, are playing more important roles as their pedigrees can be well controlled and their generation period are shorter than human. Traditional genetic crosses for identifying genetic effects typically involve crossing two inbred strains for two generations. These crosses are powerful for detecting the presence of quantitative trait loci (QTL), that is, genetic variants influencing a quantitative outcome. However, they are limited in at least two important respects. First, the small number of generations allows few chromosome crossovers (meioses) to accumulate; this limits the ways in w hich genetic variants are randomized among individuals such that QTL are localized to only a broad chromosomal region. Second, the small number of inbred strains used (ie, two âĂIJfoundersâĂİ), limits the number of QTL that could be detected: QTL can only be detected when the underlying genes differ between the two founders; genes invariant between the founders, but which in among other strain combinations would demonstrate a large effect, are hidden from the researcher.

Therefore, multiparent crosses — populations of model organisms derived from more than 2 founders and bred for more than 2 generations — are proposed to improve the resolution and power of QTL mapping on model organisms. Several multiparent crosses in mice, abrabidopsis, and maize are becoming well established. The Collaborative Cross (CC) ("Collaborative Cross Consortium", 2012) is an emerging panel of recombinant inbred mouse strains derived from 8 genetically diverse laboratory inbred strains; the Multi-parent Advanced Generation Inter-cross (MAGIC) (Cavanagh et al., 2008; Kover et al., 2009), descended from 19 founder lines of *Arabidopsis*, is a similar resource and breeding paradigm for plants. Different from CC and MAGIC, which focus on ease of replicability, Heterogeneous Stocks (HS) (Valdar et al., 2006; "Rat Genome Sequencing and Mapping Consortium", 2013) and the Diversity Outbred (Svenson et al., 2012) population aim to deliver increased mapping resolution: Through additional and continued outbreeding, their genomes are an ultra-fine-grain mosaic of the ancestral strains, collectively resembling more a population in the wild, with often complex genetic relationships between mice and a rich constellation of heterozygous and homozygous genotypic combinations. Many studies have demonstrated that the multiparent lines are very useful resources for studying complex traits by using existing general purpose tools (Aylor et al., 2011a), and several tools are developed to accommodate data analysis on multiparent lines (Liu et al., 2010; Lenarcic et al., 2012). However, since multiparent lines are recently developed, current methodology development has not been able to take full advantages of their unique characteristics, preventing the study of multiparent lines from reaching the maximal power.

Conversely, how genetic factors play in inheritance is an unimaginably complicated and intricate biological process. Since we are still in a relatively early age for studying genetics, following the principle of Occam's razor, researchers tend to directly find connections between genetic factors and observable traits without considering the intermediate biological processes. Though this principle brings in much simplicity in methodology developments, finding the potential genetic factors that influence a specific characteristic is still like looking for a needle in a haystack: a typical mammalian genome contains billions of nucleotides, in a single mammalian cell, about 360,000

RNAs are made up from 12,000 different transcripts using the DNA as the templates, and there may be trillions of cells in a single mammalian body, contributing to a myriad of quantitative and qualitative traits.

Furthermore, when investigators want to analyze the biological data from multiparent crosses, the need of the computational methodology development becomes an urgent issue: Not only should the developing methods exploit the power of multiparent crosses, generating more accurate and meaningful results than current approaches do, but also they should be able to process the massive data from the multiparent crosses, without hampering the efficiency of the study.

In addition, lots of genetics problems are profoundly related with each other. For example, RNA-Seq assembly and quantification is an innovative way to collect enormous information at transcript level, providing unique insights about the behavior of different genes in the cells. These novel traits from RNA-Seq are linked to Quantitative Trait Locus (QTLs) — DNA variants (commonly SNPs) that underline a quantitative trait (phenotype) — in QTL studies by finding all significant associations between DNA variants and the traits. The success of the gene expression based QTL studies, associating the gene expressions with the genetic variants and analyzing regulatory effects for every gene, heavily relies upon the quality of the results from RNA-Seq assembly and quantification. After the QTL study, an important question in multiparent crosses is to estimate the effects of each individual founder to the phenotype at every QTL. If the QTLs reported from the QTL studies are flawed, it is meaningless to estimate the founders' effects at QTLs, and many efforts may go in vein.

In order to alleviate the demand of new methodologies and enhance the computational pipeline for analyzing multiparent crosses, this thesis develops a series of methods for multiparent crosses to address three important computational problems.

- **RNA-Seq assembly and quantification** Current sequencing technology such as RNA-Seq enables researchers to measure transcriptome data with unprecedentedly high resolution and deep coverage (Ozsolak and Milos, 2010). However, current methods are often inaccurate, leading researchers to false conclusions (Kleinman and Majewski, 2012; van Bakel et al., 2010), and inefficient, requiring weeks if not months to finish the analysis (Patro et al., 2013).
- **Phylogeny-based QTL mapping** Though phylogeny-based QTL mapping has shown its obvious advantages over other alternative approaches in QTL mapping for inbred populations Mailund et al. (2006); Besenbacher et al. (2009); Pan et al. (2008, 2009), they cannot be easily used for analyzing multiparent crosses, since most of the multiparent crosses are either outbred or not fully inbred, containing heterozygous sites in the genome, causing multiple leaves in the final phylogenetic tree present the same sample, and weakening the power of the statistical test on effects.
- **Founder Effect Estimation** One major difference between multiparent crosses and human population is that the founders of multiparent lines are known, and this unique feature enables the possibility to estimate the founder effects at the QTL position. However, the distributions of the founders' genetic factors in the descendants are not observable, and a haplotype reconstruction step need to be applied first, providing a probabilistic way to recover the distribution of founders genomes in the descendants. But few of the existing methods properly use the probabilistic information, failing to provide statistically valid estimation on the genetic effects of founders.

Therefore, in this dissertation, a series of methods has been developed to address these problems. They are motivated by analyzing multiparent crosses, are primarily designed for multiparent crosses, and utilize the special structure and other useful information only existing in the multiparent crosses. By embedding these methods in the analysis of multiparent crosses, the trustworthiness of the results is enhanced, the computational resources used in the study are reduced, and thus, the overall competence of multiparent crosses are also improved. The following sections elaborate the problems and challenges and provide a brief overview of these methods. Here, I briefly overview my contributions to address these challenges.

1.1 RNA-Seq Assembly and Quantification and Challenges

RNA-Seq is a novel technology that allows researchers to explore the process of the transcription at resolution. In a cell, transcripts in the genome are translated to proteins. Current RNA-Seq is not able to read the whole RNA molecule, but only up to several hundredss base-pairs (nucleotides) from both ends. In order to sequence the whole molecule, a shot gun process is employed to fragment RNA molecules into short fragments, whose lengths vary from 100 base-pairs to 1000 base-pairs. And the sequences of the two ends of such fragments are read by RNA-Seq and recorded into data storage. Since different transcripts are transcribed into different number of molecules, aka, abundance levels, and for the same transcript, its abundance level varies across tissues of the same individual. Therefore, the abundance levels of the transcripts are of high interest, and it is typically considered as the RNA-Seq alternative of gene expressions.

Three sequential steps are commonly used to analyze an RNA-seq data: alignment, assembly quantification: An alignment step is required to align the reads back to the genome sequences; RNA-Seq assembly is a computational process to recover the original sequences of transcripts in the RNA-Seq data, clustering the reads based on the genome location reported by the alignment step in order to recover their originating transcripts; RNA-Seq quantification is a computational process to estimate the abundance level of transcripts in the RNA-Seq data.

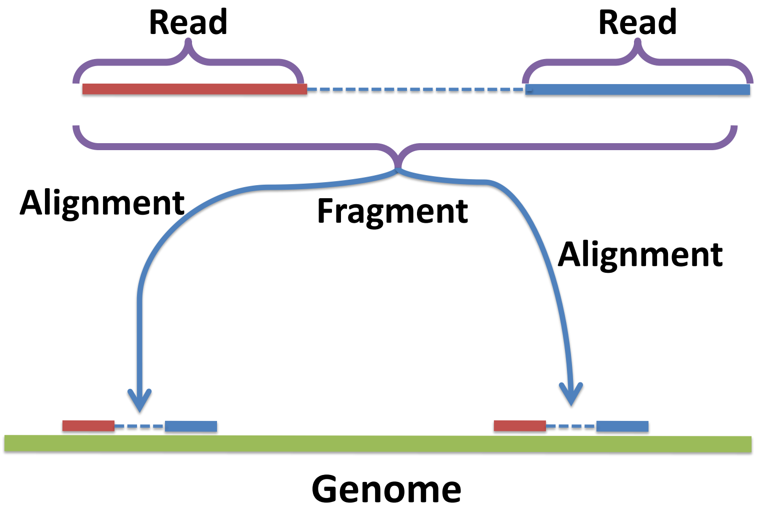


Figure 1.1: A fragment with paired end reads that can be aligned to two locations in the genome.

In this thesis, I assume that the RNA-Seq data are paired-end reads, which are widely used for transcriptome inference. Our approach can be used for single-end reads as well. In paired-end RNA-Seq data, a fragment is a sub-sequence from an expressed transcript. High-throughput sequencing provides two reads corresponding to the two ends of the fragment. In this thesis, two primary computational challenges in RNA-Seq are addressed.

First, if a fragment can be mapped to more than one locations in the genome, this fragment has *multiple alignments*, as showed in Figure 1.1. Since each fragment originates from only one location in the genome, multiple alignments must be processed/-corrected before subsequent analysis can proceed. However, some multiple alignments from the repetitive regions on the genome are unidentifiable because their sequences are exactly the same. Without extra information from other sources, there is no way to correct this type of multiple alignments. Multiple alignments are a major source of false positives in RNA assembly (Kleinman and Majewski, 2012; van Bakel et al., 2010).

In this dissertation, I examine the underlying genomic features that lead to multiple alignments and investigate how they generate systematic errors in RNA-Seq analysis. GeneScissors is developed, exploiting machine learning techniques guided by biological knowledge to detect and correct spurious transcriptome inference by existing RNA-Seq analysis methods. In the simulated study, GeneScissors can predict spurious transcriptome calls due to misalignment with accuracy close to 90%. It provides substantial improvement over the widely used TopHat/Cufflinks or MapSplice/Cufflinks pipelines in both precision and F-measurement. On real data, GeneScissors reports 57.6% less pseudogenes and 0.95% more expressed and annotated transcripts, when compared with the TopHat/Cufflinks pipeline. In addition, GeneScissors finds that more than 18% of the unannotated genes reported by the TopHat/Cufflinks pipeline are false positives.

Second, though the alignment step is a critical step to help RNA-Seq assembly to detect novel transcripts, it has become a bottleneck for the RNA-Seq quantification due to its long running time for exhaustively searching every possible splice junctions in the reads. To address this problem, others have started to develop alternative algorithms to conduct RNA-Seq quantification without alignments (Sailfish (Patro et al., 2013)).

In this thesis, I proposed a novel RNA-Seq quantification method, RNA-Skim, which partitions the transcriptome into disjoint transcript clusters based on sequence similarity and introduces the notion of sig-mers that are special k-mers uniquely associated with each cluster of transcripts. More importantly, unlike other approaches, RNA-Skim does not depend on a time-consuming RNA-Seq aligner. I demonstrate that the sig-mer counts within a cluster are sufficient for estimating transcript abundances with accuracy comparable to any state of the art method. This enables RNA-Skim to perform transcript quantification on each cluster independently, reducing a complex optimization problem into smaller optimization tasks that can be run in parallel. As a result, RNA-Skim uses less than 4% of the k-mers and less than 10% of the CPU time required by Sailfish. It is able to finish transcriptome quantification in less than 10 minutes per sample by using just a single thread on a commodity computer, which represents more than 100 times speedup over the state of the art alignment-dependent methods, while delivering comparable or higher accuracy.

1.2 QTL mapping and Challenges

The goal of Quantitative Trait Locus (QTL) mapping is to find strong associations representing (genomically proximal) causal genetic effects between observed quantitative traits and genetic variations. The founders of multiparent crosses are typically chosen to maximize some criterion of genetic diversity, and the breeding scheme is typically designed to maintain that diversity while fractionating the genome into small haplotype segments. The resulting multiparent crosses, comprising individuals whose genomes are fine-grain mosaics of the original founders, is well suited to detection of quantitative trait loci (QTL) through linkage mapping.

The most common genetic variations used in QTL mapping are single nucleotide polymorphisms (SNPs). A SNP is a single nucleotide difference in the same location of different DNA sequences, e.g., the DNA sequences in different haploid or diploid individuals, or the pair of the DNA sequences in the same diploid individual. In theory, a SNP can have up to four different variants, including all four nucleotides: A, T, C, G. In reality, most of SNPs only show variation between two out of four possible ones. So, SNPs are usually encoded in a binary presentation, and standard approaches for QTL mapping is to find the SNPs whose binary representation have the strongest correlations with quantitative traits.

Several approaches have been proposed, e.g., single markers based (Akey et al., 2001; Pe'er et al., 2006; Thomas, 2004), haplotype-based (McClurg et al., 2006; Onkamo et al., 2002; Li and Jiang, 2005) , phylogeny-based methods (Mailund et al., 2006; Besenbacher et al., 2009; Pan et al., 2008, 2009). Among these methods, local phylogeny based QTL mapping is a popular method to discover the significant association between each of the regions segmented by local phylogeny trees and the phenotypes. However, they can be only directly applied for haplotypes from inbred populations or haploid populations, otherwise, a *phasing* step, which is time-consuming and error prone, is required. This obstructs the efficiency of local phylogeny based QTL mapping. In this thesis, I explore the possibility to extend local phylogeny based QTL mapping to the population with heterozygous sites.

A new method, HTreeQA, is proposed to use a tri-state semi-perfect phylogeny tree to approximate the perfect phylogeny used by existing methods. The semi-perfect phylogeny trees are used as high-level markers for association study. HTreeQA uses the genotype data as direct input without any phasing step, and it can handle complex local population structures. It is suitable for QTL mapping on any multiparent crosses. Simulation studies under three different genetic models show that HTreeQA can detect a wider range of genetic effects and is more efficient than existing phylogeny-based approaches. QTLs are also found for two phenotypes of the incipient of Collaborative Cross, which are consistent with known genes and QTL discovered in independent studies.

1.3 Haplotype Effects Estimation and Challenges

QTL mapping in multiparent crosses is a powerful approach for investigating the genetic basis of variation in complex traits, and in particular those methods focusing on detection of single QTL, are relatively well established. But just finding QTLs is not enough. Methods to characterize QTL effects those estimating how inheritance of alternate founder haplotypes drives phenotypic outcome remain in their infancy.

Because the genome of each individual in multiparent population can be described as a mosaic of founder haplotypes, any given point in that genome can likewise be described in terms of the pair of haplotypes (ie, *diplotype*) present. Although the identity of this diplotype in most cases cannot be observed directly, it can be probabilistically inferred from genotype data. A number of algorithms have been developed to do this, notably those based on a hidden Markov model (HMM) formulation (eg, HAPPY, (Mott et al., 2000); GAIN (Liu et al., 2010)). In the HMM framework, diplotypes are modeled as latent outcomes drawn from a discrete set of possibilities; genotype data provides partial information about this underlying latent state, and so the HMM's reconstruction of the haplotype mosaic leads to haplotype assignments that are probabilistic — a list of probabilities for each possible diplotype state at each locus for each individual. Despite the fact that haplotype composition is itself uncertain, the estimation of haplotype (founder) effects must proceed because it is vital for understanding the underlying mechanism of how different genetics factors from founders contribute the QTL.

A general Bayesian framework, Diploffect, is described for estimating the effects of founder haplotypes at quantitative trait loci detected in multiparental genetic populations. The aim is to provide a framework for coherent estimation of haplotype and diplotype (haplotype pair) effects that takes into account: uncertainty in the haplotype assignments for each individual; uncertainty arising from small sample sizes and infrequently observed haplotype combinations; possible effects of dominance (for non-inbred subjects); kinship effects (for population structure); and that provides a means to incorporate data that may be incomplete or that has a hierarchical structure. Different from existing methods, Diploffect uses the results of a probabilistic haplotype reconstruction as prior information to obtain posterior distributions at the QTL for both haplotype effects and haplotype composition. Two alternative computational approaches are studied: a Markov chain Monte Carlo sampler, and an Importance Sampling procedure. Results are presented for quantitative phenotypes in simulated CC and HS populations, and both quantitative and binary phenotypes in incipient CC and HS. Compared with existing approaches, Diploffect produces not only more robust point estimates of diplotype effects but also — essential for prioritizing follow-up experiments — confidence (credibility) intervals that are statistically valid and allow effects of different haplotypes to be meaningfully compared.

1.4 Thesis Statement

- Current methods for quantifying relative transcript abundance from RNA-Seq are computationally demanding. Efficiency could be improved by using algorithms that better exploit redundancy in the data.
- Methods for analyzing mulitiparent crosses would be more powerful when they incorporate existing biological knowledge and the unique structure of multiparent crosses.

1.5 Thesis Outline

This thesis is organized as follows:

- The GeneScissors method is presented in Chapter 2.
- The RNA-Skim method is presented in Chapter 3.
- The HTreeQA method is presented in Chapter 4.
- The Diploffect method is presented in Chapter 5.
- Chapter 6 concludes my thesis work and outlines the future work.

CHAPTER 2

GENESCISSORS: A COMPREHENSIVE APPROACH TO DETECTING AND CORRECTING SPURIOUS TRANSCRIPTOME INFERENCE DUE TO RNASEQ READS MISALIGNMENT.

RNA-Seq techniques provide an efficient means for measuring transcriptome data with high resolution and deep coverage (Ozsolak and Milos, 2010). Millions of short reads sequenced from cDNA provide unique insights into a transcriptome at the nucleotide-level, and mitigate many of the limitations of microarray data. Although there are still many remaining unsolved problems, new discoveries based on RNA-Seq analysis ranging from genomic imprinting (Gregg et al., 2010) to differential expression (Trapnell et al., 2012a; Anders and Huber, 2010a) promise an exciting future.

Though RNA-Seq sequencing offers an alternative to Microarray techniques in gene expression study (Wang et al., 2009), it also brings new challenges, including how to detect and assembly novel transcripts in the data, how to rapidly and effectively process the massive data produced by the proliferation of RNA-Seq high-throughput sequencing, how to build statistical model for accurate quantification of transcript abundances for transcriptome, etc.

In the next two chapters, I present two different RNA-Seq methods, representing another two different ways to use the genome databases associated with multiparent lines in RNA-Seq analysis. Modern RNA-Seq quantification methods are categorized into two major groups: alignment-dependent and alignment-free methods. This chapter details GeneScissors that utilizes the machine learning based method to correct multiple alignment errors and remove misclassified genes reported by the current RNA-Seq tools such as Cufflinks. This chapter primarily focuses on arising from the alignment-dependent methods. Chapter 3 a novel method for RNA-Seq, but shifts the focus to the alignment-free tools instead. It is also worth mentioning that these two RNA-Seq methods can also be applied to populations other than multiparent lines, as long as the DNA genome or list of variants of the samples are available (e.g. the results of DNA-Seq assembly from the samples).

2.1 Introduction

Current RNA-Seq alignment-based pipelines typically contain two major components: an aligner and an assembler. An RNA-Seq aligner (e.g. TopHat (Trapnell et al., 2009), SpliceMap (Au et al., 2010), MapSplice (Wang et al., 2010)) attempts to determine where in the genome a given sequence comes from. An assembler (e.g. Cufflinks (Trapnell et al., 2010), Scripture (Guttman et al., 2010)) addresses the problems of which transcripts are present and estimating their abundances.

Existing alignment-based pipelines can be further divided into two major categories: align-first pipelines and assembly-first pipelines (Ozsolak and Milos, 2010). Assembly-first pipelines attempt to assemble and quantify the complete transcriptome without a reference. Several algorithms, such as Trinity (Grabherr et al., 2011a) and TransABySS (Robertson et al., 2010), have been developed. However, aligning fragments to a reference genome is still necessary in order to interpret the results from an assembly-first pipeline and to relate them to existing knowledge. The assembly-first pipeline is computationally intensive, requiring several days to complete. In align-first pipelines a high-quality reference genome serves as a scaffold for inferring the source of RNA-Seq fragments. Current alignment approaches are both computationally more efficient and easier to parallelize than assembly-first pipelines. Thus, the align-first RNA-Seq analysis can be finished within hours even on a normal desktop machine.

Therefore, align-first pipelines such as TopHat/Cufflinks (Trapnell et al., 2010, 2012a) or MapSplice/Cufflinks (Wang et al., 2010) are generally preferred when a suitable reference genome is available.

2.1.1 Multiple-Alignment Problem

If a fragment can be mapped to more than one location in the genome, this fragment has *multiple alignments*, as showed in Figure 1.1. Since each fragment originates from one location in the genome, multiple alignments must be processed/corrected before subsequent analysis can proceed. Inappropriate handling of the multiple alignment fragments impacts the subsequent analysis and may lead to questionable conclusions. For example, the “widespread RNA and DNA sequence differences” (Li et al., 2011) are suspected to be (at least partially) due to systematic technical errors, including misalignments (Kleinman and Majewski, 2012).

Current RNA-Seq analysis pipelines handle the multiple-alignment problem in both the alignment and assembly steps. Most existing aligners (e.g. TopHat (Trapnell et al., 2009)) use a scoring system where only the alignments with the “best score” are kept. However, a fragment may still have multiple alignments with equally good scores. In the experiments on real mouse RNA-Seq data, I observe that at least 5% fragments have multiple alignments. The assembler (e.g., Cufflinks (Trapnell et al., 2010)) either assumes that they contribute equally to each location or uses a probabilistic model to estimate their contributions based on the abundance of the corresponding transcripts (Li et al., 2010).

2.1.2 Genomic Factors Causing Multiple Alignments

In general, multiple alignments are caused by the existence of paralogous sequences within a genome. Duplicated and repetitive sequences need not be strictly identical. In this subsection, we discuss genomic factors that may lead to multiple alignments and their impact on RNA-Seq analysis. Retrotransposition and gene duplication are two biological phenomena that generate sequences with high levels of nucleotide similarity. Interspersed highly repetitive sequences, such LINEs and SINEs, can be expressed in an autonomous or non-autonomous manner but are not our focus. That leaves us with three major types of genomic factors: processed pseudogenes (Balakirev and Ayala, 2003; Vanin, 1985; Zhang et al., 2003), non-processed pseudogenes (Hurles, 2004), and repetitive sequences shared by gene families (Häsler et al., 2007; Jurka and Smith, 1988).

The functions of pseudogenes (Harrison et al., 2003; Khelifi et al., 2005) are still under investigation (for example, (Hirotsune et al., 2003)). Pseudogenes are generally caused by DNA duplication or RNA retrotransposon. They can be further categorized in two groups: processed pseudogenes and non-processed pseudogenes based on their causes. Both lead to the repetitive genomic sequences. In general, lots of these pseudogenes are nonfunctional, and under reduced selection pressure, thus, they typically exhibit a higher mutation rate than the expressed genes from which they originated.

Processed pseudogene: A processed pseudogene (Vanin, 1985) is generated when an mRNA is reverse transcribed and reintegrated back to the genome. The resulting DNA sequence of the processed pseudogene is the concatenated exon sequences from its original transcript. Because there are no splice junctions in the sequence of the processed pseudogene, it is easier for the current RNA-Seq aligners to map the fragments to processed pseudogene, than the actual gene from which they are expressed, especially those fragments that cross a splice junction. Both the unexpressed pseudogene and its corresponding expressed gene may be reported by the assembler if the implementation of the assembler does not consider such cases. For example, Guttman et al., 2010 observed that a few highly expressed transcripts may not be able to be fully reconstructed due to alignment artifacts caused by the processed pseudogenes.

Nonprocessed pseudogene: Nonprocessed pseudogenes (Hurles, 2004) are typically caused by a historical gene duplication event, followed by an accumulation of mutations, and an eventual loss of function. Nonprocessed pseudogenes often share similar exon/intron structures with their originating gene. From the aligner's perspective, fragments can be mapped to either the expressed original gene, or its nonprocessed pseudogene, or both. Similar to processed pseudogenes, the assembler may report a nonprocessed pseudogene when its corresponding functional genes is expressed.

Repetitive shared sequences: Besides pseudogenes, many functional gene families share subsequences that are almost identical to each other. One repetitive sequence shared by different genes in human genome is *Alu* (Jurka and Smith, 1988; Häsler et al., 2007). Consider the case when, among all genes which share the *Alu* sequence, but only a subset are expressed. Hence the aligner will map the fragments originating from the expressed subset to all similar sequences on the genome. And the assembler may report all genes sharing the repetitive sequence as being expressed.

Any of these three biological factors may lead to multiple alignments. Without proper post-processing, an assembler may report many unexpressed pseudogenes or even random regions as expressed genes, and it may also miss a few highly expressed genes.

Existing RNA-Seq analysis pipelines provide heuristics for addressing the multiple alignment problem, however, they do not explicitly consider their genomic causes. In

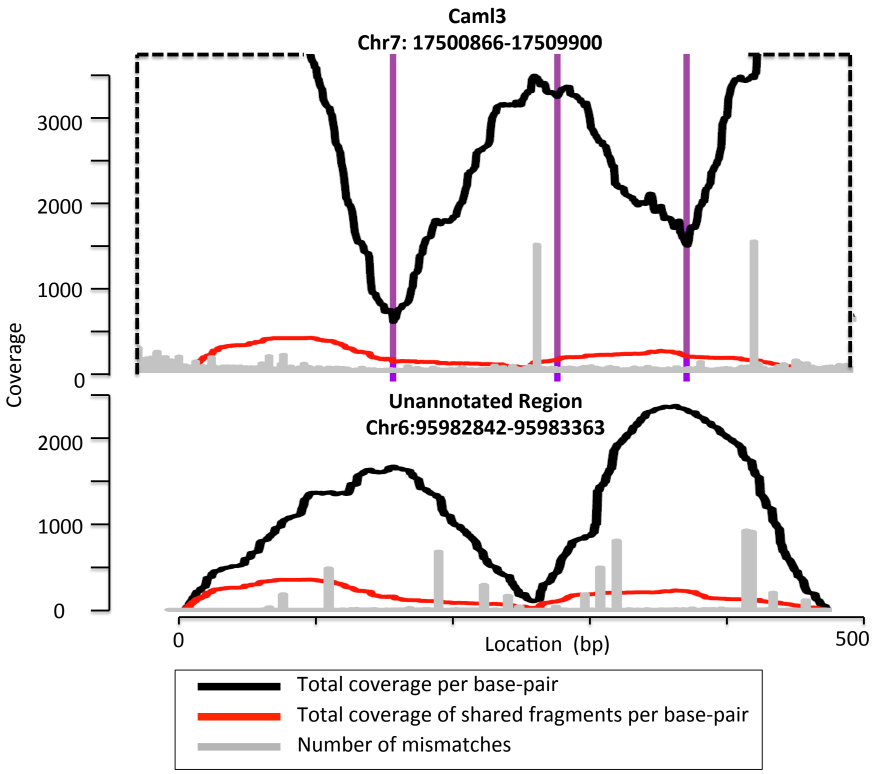


Figure 2.1: Two transcripts reported by Cufflinks. The top one maps to a known gene named *Caml3*, and the bottom one does not map to any known gene. Two transcripts are aligned by their shared fragments in the plot. The top figure is truncated, and only shows the region containing shared fragments. The dashed line indicates the truncated boundary. The three vertical lines in purple represent three splice junctions in the top transcript. The points on the black line represent the numbers of fragments that cover the corresponding base pairs. The points on the read represents the number of fragments that cover the corresponding base pairs and are also aligned to the other transcript.The gray lines represent the number of mismatches across the regions in the plot.

the study using mouse RNA-Seq data, the transcripts reported by Cufflinks include about 3.5% from known pseudogenes and about 10% from unannotated regions. A quarter of these 13.5% transcripts are likely to be false positives caused by multiple alignments.

Figure 2.1 shows the pile-up plots of two regions from a mouse genome reported by a current RNA-Seq pipeline. The top one is a gene named *Caml3*, while the bottom one is unknown. The unknown gene's sequence is very similar to the sequence of concatenated exons from *Caml3*. Fragments that are uniquely aligned to the unknown gene by the aligner can also be aligned to *Caml3*. However, the aligner fails to find the proper alignment because it does not consider all possible alignments crossing splice junctions due to the search complexity. This collection of evidence indicates that the unknown gene is actually an unannotated processed pseudogene of *Caml3*.

Therefore, the identification of expressed genes and unexpressed pseudogenes is a significant confounding factor in RNA-Seq analysis. No existing analysis methods explicitly attempt to identify and reassign fragments that are mapped to pseudogenes. A similar observation was made by ContextMap (Bonfert et al., 2012) that multiple alignments from a RNA-Seq aligner could be handled by removing the incorrect alignments based on the "context" of the alignments. However, ContextMap simply defines the "context" as a fixed window around the alignment on the genome. It also does not try to rescue any missed alignments. In contrast, I introduce the concept of fragment attractor, which leverages the results from both an aligner and an assembler to determine the appropriate "context" for each individual alignment. Sharing maps between fragment attractors are built to help discover and restore missed alignments.

In this chapter, I introduce the GeneScissors pipeline (Zhang et al., 2013), a comprehensive approach to address the problem of detecting and correcting those fragments errantly aligned to unexpressed genomic regions. When compared with the standard TopHat/Cufflinks pipeline, GeneScissors is able to remove 57% pseudogenes without using any annotation database. GeneScissors can reduce inference errors in existing analysis pipelines and aid in distinguishing truly unannotated genes from errors.

2.2 Methods

In this section, GeneScissors is presented, a general component that can be applied to any align-first RNA-Seq pipeline to detect and correct errors in transcriptome in-

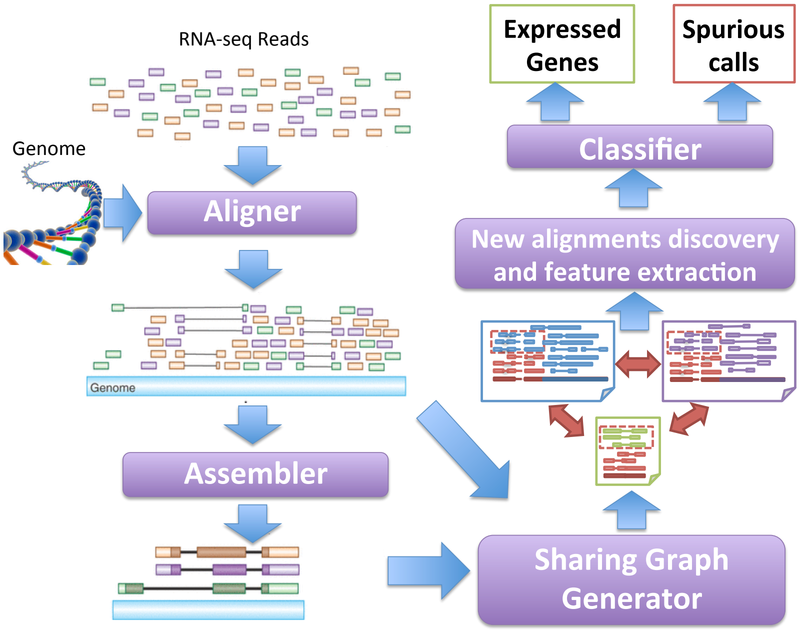
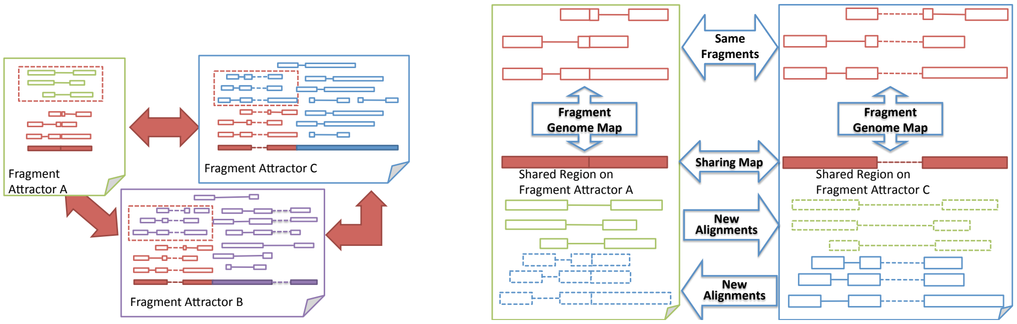


Figure 2.2: The workflow of GeneScissors Pipeline. The traditional RNA-Seq analysis pipeline is the path on the left side. Its alignment and assembly results are used by GeneScissors to infer fragment attractors, build sharing graphs, and identify all fragment alignments in the genome. GeneScissors then builds a classification model to detect and remove unexpressed genes.

ference due to fragment misalignments. In a standard RNA-Seq pipeline, the "best" alignment for a fragment with multiple alignments is determined without considering the surrounding alignments of other fragments. Such decisions may be premature without considering the other fragments aligned to these regions. GeneScissors pipeline first collects all possible alignments for all fragments, and then examines those regions of the genome where multiple alignments map and then consider the other fragments aligned to these regions. In this way, GeneScissors is able to leverage statistics of fragment distribution and other features of the alignments.

Figure 2.2 describes the proposed workflow for RNA-Seq analysis. It utilizes existing aligner and assembler (with minor modifications to keep all possible alignments discovered, details in Section 2.3.1) to identify regions to which fragments align. In



(a) (b)

Figure 2.3: Figure 2.3(a) shows a sharing graph of three fragment attractors A, B, and C. Each solid box represents a pile-up of fragments of a fragment attractor. Each pair of connected hollow rectangles represents a fragment of paired-end reads. The red fragments are the shared fragments that can be mapped by the aligner to all three fragment attractors. The bottom row in each box represents the transcript sequence. The red regions (except the splice junctions in the transcript sequences) are the region to which the shared fragments align. Figure 2.3(b) shows a sharing map between fragment attractors A and C and the discovered new alignments (shown in dashed rectangles). These new alignments are rescued from the uniquely aligned fragments in the shared region of one of the two fragment attractors.

order to distinguish from expressed genes, each such region is referred as a *fragment attractor*. Fragments with multiple alignments *link* corresponding fragment attractors. these fragments and their alignments are referred as *shared fragments* and *shared alignments* respectively. The relationships among linked fragment attractors are defined by their *shared fragments*. GeneScissors uses *sharing graphs* to represent the linked fragment attractors and to discover new fragment alignments. Training instances are created by using simulated RNAseq fragments from annotated genes in Ensembl to build a classification model. Then, on real data, the classification model predicts and removes the fragment attractors that are likely due to misalignments. Existing assembly methods can be applied on the remaining fragment alignments to re-estimate the abundance level of expressed fragment attractors. The sharing graph is introduced in Section 2.2.1, a classification model to identify the unexpressed fragment attractors in Section 2.2.2 and the features extraction method from the sharing graphs in Section 2.2.3.

2.2.1 Sharing Graph

Sharing graphs are constructed as follows. Each fragment attractor is represented by a node and each pair of linked fragment attractors are connected by an edge. Each connected component is called a *sharing graph*. For each edge in a sharing graph, a position-by-position *sharing map* is built between the pair of linked fragment attractors through their shared fragments. For any fragment f aligned to a fragment attractor g, function $\phi_{f \Rightarrow g}$ is defined, which returns the aligned position in fragment attractor g given a position in fragment f, and its inverse function $\phi^{-1}_{g \Rightarrow f}$, which returns the corresponding position in f (if it exists) given a position in g. For a pair of linked fragment attractors g_a and g_b and one of their shared fragments f_1, position k in f_1 may be aligned to position i in fragment attractor g_a and position j in g_b. This provides a correspondence between position i in g_a and position j in g_b by $j = \phi_{f_1 \Rightarrow g_b}(\phi^{-1}_{g_a \Rightarrow f_1}(i))$ and $i = \phi_{f_1 \Rightarrow g_a}(\phi^{-1}_{g_b \Rightarrow f_1}(j))$. A sharing map can be built between g_a and g_b through this approach by using all their shared fragments. It is possible that two shared fragments f_1 and f_2 map the same position in g_a to two different positions in g_b, i.e., $\phi_{f_1 \Rightarrow g_b}(\phi^{-1}_{g_a \Rightarrow f_1}(i)) \neq \phi_{f_2 \Rightarrow g_b}(\phi^{-1}_{g_a \Rightarrow f_2}(i))$. Empirically, such cases are rare, and when it happens, the majority rule is used to resolve the conflict.

The region of a fragment attractor that is covered by the sharing map is called the *shared region*. In addition to the shared fragments, some other fragments uniquely aligned to the fragment attractor may align to the shared region. These fragments should have been aligned to the linked fragment attractor too, but the aligner might have failed to recognize the alignments due the reasons discussed previously. Therefore, with the help of the sharing map, ' these missed alignments can be restored from existing aligner's result. For example, in Figure 2.3(a), a sharing graph among three fragment attractors are shown. The red regions in the bottom row of each fragment attractor are the shared regions. The red dashed boxes contain the fragments uniquely aligned to one fragment attractor by the aligner, but should have been aligned to the linked fragment attractors too. In Figure 2.3(b), more details are shown on how the new alignments of the fragments are established through the sharing map. Note that this alignment discovery operation needs to be done in both directions for each pair of linked fragment attractors. In the previous example in Figure 2.1, the uniquely aligned fragments (between the black curve and the red curve) in the shared regions should have been aligned to both fragment attractors. Restoring fragment alignments to multiple positions does not cause inflation in abundance level estimation because transcriptome inference methods such as Cufflinks already consider the shared alignments. This approach enables us to safely rescue fragment alignments missed by an aligner.

2.2.2 Classification Model

GeneScissors processes RNA-Seq data at the granularity of linked fragment attractors. Because there is no easy way to determine whether a fragment attractor are expressed or not in real datasets, I build the training model from simulated data and apply it to real data. I first generate our training set from a simulated population, and each sample is a set of fragments simulated based on a set of selected transcripts from the annotation database. (More details are in Section 2.3.1). Then, the aligner and the assembler are applied on each sample of the simulated data, the sharing graphs are built based on their results, and training instances are also generated based on the sharing graphs. The fragment attractors which cannot be mapped back to the selected transcripts are unexpressed ones. A classification model is used to infer whether a fragment attractor (hereby referred to as the *target* fragment attractor g_t) is expressed or not using features of g_t and another fragment attractor (hereby referred to as the *assistant* fragment attractor g_a) linked to g_t by an edge in the sharing graph. For every pair of linked fragment attractors, two instances are built. The instance is labeled according to whether the target fragment attractor is expressed. Therefore, one fragment attractor may be the target fragment attractor in multiple instances. The intuition is that, for an unexpressed target fragment attractor, there should always be some instances in which the assistant fragment attractors are expressed. In such instances, the assistant fragment attractor should have less consistent mismatches, longer sequence and lower proportion of shared fragments than the target fragment attractor (More details are in Section 2.2.3 which describes all features used in this Chapter). Thus a binary classification model can be training, using these features to identify unexpressed target fragment attractors. When the model is applied to test data and real data, all target fragment attractors which are predicted as unexpressed at least once will be removed from the result of the assembler, and the reads that are uniquely aligned to these fragment attractors will be redistributed to the corresponding expressed fragment attractors. I experimented with SVM, DecisionTrees, and RandomForests as the learning method, and found that RandomForests had the best overall performance. Once the classifier is built, I apply it on test data to evaluate the prediction accuracy and then apply it to real data to predict *unexpressed* fragment attractors and remove their fragment alignments. Recall that, for all uniquely aligned fragments in the shared regions of these fragment attractors, new alignments can also be discovered to their linked fragment attractors using the sharing map.

2.2.3 Fragment Attractor Features

Features are extracted from both target fragment attractor g_t and assistant fragment attractor g_a in each instance. Each instance contains 14 features, listed in Table 2.1. All features except the number of consistent mismatch locations (details later) are straightforwardly calculated: features NE and NI are directly collected from the assembler's output, and NR, MF, MR and CM are calculated by our sharing graph generator. The

Features	Description
$\mathrm{NE}(g_a) == 1, \mathrm{NE}(g_t) == 1$	$\mathrm{NE}(g_a)$ and $\mathrm{NE}(g_t)$ are the observed numbers of exons. These two Boolean features tell whether the genes are singleton of exons or not.
$\mathrm{NR}(g_a)$, $\mathrm{NR}(g_t)$, $\mathrm{NR}(g_a)/\mathrm{NR}(g_t)$	$\mathrm{NR}(g_a)$, $\mathrm{NR}(g_t)$ are the proportions of the fragments that can be aligned to g_a and g_t to the total fragments, respectively.
$\mathrm{MF}(g_a)$, $\mathrm{MF}(g_t)$, $\mathrm{MF}(g_a)/\mathrm{MF}(g_t)$	$\mathrm{MF}(g_a)$, $\mathrm{MF}(g_t)$ are the proportions of the shared fragments to the fragments aligned g_a and g_t, respectively.
$\mathrm{MR}(g_a)$, $\mathrm{MR}(g_t)$, $\mathrm{MR}(g_a)/\mathrm{MR}(g_t)$	$\mathrm{MR}(g_a)$, $\mathrm{MR}(g_t)$ are the proportions of the entire regions of g_a and g_t that are covered by shared fragments.
$\mathrm{CM}(g_a)$, $\mathrm{CM}(g_t)$, $\mathrm{CM}(g_a) - \mathrm{CM}(g_t)$	$\mathrm{CM}(g_a)$, $\mathrm{CM}(g_t)$ are the numbers of base pairs that have consistent mismatches in the shared regions of g_a and g_t respectively.

Table 2.1: The features used for detecting fragment attractors resulting from misalignments.

use of consistent mismatch count CM as a feature is motivated by the observation that the pseudogenes usually have higher mutation rate. The number of exons are helpful in distinguishing processed pseudogenes, which are singletons. All the other features are motivated by our observation that the unexpressed fragment attractors tend to have smaller number of alignment fragment and shorter region than their corresponding expressed ones.

2.2.4 Consistent Mismatches Discovery in GeneScissors

In this section, I describe the concept of consistent mismatch and the method to find consistent mismatch locations across the genome.

For a given base pair location in the genome, if the number of aligned fragments that carry an allele different from the reference genome is much higher than the expected number due to random sequencing errors, it is called as a *consistent mismatch* location. There are three possible reasons that a consistent mismatch occurs: 1) A missing SNP or heterozygous site in a diploid sample's genome (inconsistency between the reference DNA sequence and the sample's DNA sequences), 2) an RNA-editing site, and 3) misaligned fragments (difference between the sequences of a gene and its pseudogene). Consider the example shown in Figure 2.1, there are two visible consistent mismatches on the expressed gene, *Caml3*, and they are due to either of the first two reasons (an unreported SNP, a heterozygous SNP, or an RNA-editing event). Because the fragments aligned to the unannotated region originated from *Caml3*, in the pile-up plot of the unannotated region, there are more than six visible consistent mismatches due to the third reason (misaligned fragments).

It is important to separate the consistent mismatches from the mismatches due to sequencing errors. I assume that the sequencing error rate of a given base pair c in a given fragment is reflected in its quality score q_c, and can be derived as a function $e(q_c)$. Given a base-pair location l in the genome, let $R(l)$ be the set of base-pairs aligned to the location. The number of mismatches $\mathrm{NM}(l)$ at this location should follow a sum of Bernoulli distributions with different success probabilities, which is $M = \sum_{c \in R(l)} \mathrm{Bernoulli}(e(q_c))$. The p-value of the location is defined as $P(M \geq \mathrm{NM}(l))$. A significant p-value indicates that this location may be a consistent mismatch location. In order to find all consistent mismatch locations, the first thing is to estimate the sequencing error rate. The original function to calculate the error rate is

$$e(q_c) = \frac{\text{Total number of mismatches occurring with quality } \mathrm{q_c}}{\text{Total number of base pairs with the quality } \mathrm{q_c}}.$$

In this calculation, the consistent mismatches should be excluded as they are not caused by sequencing errors. This can be done iteratively, starting from an initial estimation using all positions that have at least ten fragments aligned. In each iteration, if the positions on the genome have much higher mismatch rate than the current estimated error rate and re-estimate the error rate, they are masked as consistent mismatches. The empirical distribution of e is the new estimation of e. For the positions that contain less than three mismatches, the computation of following two probabilities are in $O(|R(l)|)$ time complexity :

$$P(M=0) \;=\; \prod_{c\in R(l)} (1-e(q_c)) \tag{2.1}$$

$$P(M=1) \;=\; \sum_{c\in R(l)} e(q_c) \prod_{c'\in R(l)/\{c\}} (1-e(q_{c'})) \tag{2.2}$$

then the exact probability as the p-value is calculated in the following way:

$$P(M\geq 0) \;=\; 1 \tag{2.3}$$

$$P(M\geq 1) \;=\; 1-P(M=0) \tag{2.4}$$

$$P(M\geq 2) \;=\; 1-P(M=0)-P(M=1) \tag{2.5}$$

The number of mismatches at a position should distribute as a sum of a series of random variables from Bernoulli distributions with different parameters, and the distribution of the sum can be approximated by a Poisson distribution based on Le Cam's theorem (Le Cam, 1960):

$$\mathrm{CM}(l) = \sum_{c\in R(l)} \mathrm{Bernoulli}(e(q_c)) \approx \mathrm{Poisson}(\sum_{c\in R(l)} e(q_c)),$$

where $\mathrm{CM}(l)$ is the number of consistent mismatches at the location l. Therefore, the p-value can be approximated by

$$P(M\geq \mathrm{CM}(l)) \approx \sum_{m>=\mathrm{CM}(l)} f_{\mathrm{Poisson}}(m;\sum_{c\in R(l)} e(q_c)),$$

where f_{Poisson} is the density function for a Poisson distribution. The positions with p-values less than 10^{-20} are classified as consistent mismatch locations. This process continues until no more consistent mismatch locations are found. This threshold is empirically determined because current threshold gives us the best performance to identify the unexpressed genes.

2.3 Results

I first describe a series of modifications made to open-source RNA-Seq analysis tools to support GeneScissors. Then I describe the various datasets used for evaluation. I evaluated two standard pipelines that do not use GeneScissors: one using TopHat and the second using MapSplice as an aligner. Then GeneScissors are added to each pipeline, to improve the alignment results, and they are referred as GeneScissors (TopHat) and GeneScissors (MapSplice) pipelines. All four pipelines use Cufflinks as the transcriptome assembler.

2.3.1 Software

GeneScissors uses modified versions of TopHat and Cufflinks, and employs components written in C++, Python, and the BamTools (Barnett et al., 2011) library. Cuffcompare is used to map the reported genes back to Ensembl annotations, and categorize them into three types: annotated normal genes/transcripts, annotated pseudogenes, and unannotated regions.

Modifications to TopHat and Cufflinks I first present the algorithms used by TopHat and Cufflinks in ranking and reporting alignments and genes, and then discuss our modifications to retain all fragment and partial fragment (unpaired reads) alignments.

In TopHat, if the fragment f has multiple alignments x and y, TopHat retains only alignment y and does not report alignment x, when one of the following conditions is satisfied (tests are applied in order) :

- **Mismatch rule:** x has more mismatches than y.

- **Splice junction rule:** x crosses more splice junctions than y.
- **Other rules:** I omit the conditions that are not relevant to the method.

Only alignments with the best score are reported by TopHat. One observation is that the splice-junction rule tends to favor processed pseudogenes; the correct alignment of a fragment with a splice junction is frequently discarded by TopHat if the fragment can be aligned to a processed pseudogene with the same number of mismatches.

In Cufflinks, a gene that meets the following criteria is suppressed:

- 75% **rule**: More than 75% of the fragment alignments supporting the gene are mappable to multiple genomic loci.

Consider the example shown in Figure 2.1. Cufflinks fails to remove the unannotated pseudogene, which is composed mostly of uniquely aligned fragments. This suggests that the 75% rule is insufficient.

Therefore, in the GeneScissors pipeline, the splice junction rule is disabled in TopHat and the 75% rule in Cufflinks.

Simulator In order to generate training data for our classification model and evaluate the effectiveness of GeneScissors for detecting and removing unexpressed fragment attractors, a RNA-Seq simulator is used to provide a “ground truth” model for fragment attractors. The simulator randomly chooses a (user-specified) number of genes, and for each gene, it samples a subset of its transcripts. Then, it uniformly samples paired-end fragments up to a certain abundance level for each selected transcript. For each fragment, it assigns a quality score to each base pair, drawing from an empirical distribution derived from real data, and generates base-pair errors based on their quality scores.

2.3.2 Materials

The study used inbred and F1 crosses of three mouse strains: CAST/EiJ, PWK/PhJ, and WSB/EiJ. In order to minimize the impact of unknown SNPs to the alignments, strain-specific genomes are generated by incorporating high-confidence SNPs detected in a recent DNA sequencing project of laboratory mouse strains conducted by the Welcome Trust (Keane et al., 2011) into the *mm9* reference genome. The Ensembl database (build 63) (Flicek et al., 2011) is used to annotate and evaluate the results from real and simulated data.

Simulated Data A RNA-Seq simulator was used to generate synthetic data from 60 RNA-Seq samples also derived from three inbred mouse strains: CAST/EiJ, PWK/PhJ, and WSB/EiJ. In each sample, $13,000$ annotated functional genes are selected in Ensembl as the expressed genes, and randomly set them to different levels of abundance. Note that many genes included multiple transcripts. generated 10 million fragments with 100 base-pair, paired-end reads for each sample are also generated. TopHat and MapSplice as aligners and Cufflinks as the assembler are used to analyze the simulated data. More than 7.5$\%$ of the genes reported in the results were not from the selected genes in our simulation setting. From the results, shared graphs are built and the model is trained and tested by cross-validation. A feature selection study using the simulated data can be found in the supplementary material.

Real data GeneScissors is applied to RNA-Seq data from 9 inbred samples and 53 F1 samples derived from three inbred mouse strains CAST/EiJ, PWK/PhJ, and WSB/EiJ. cDNA are sequenced from mRNA extracted from brain tissues of 3-6 replicates of both sexes and the 6 possible crosses (including the reciprocal). To mitigate misalignment errors due to heterozygosity, for each F1 sample, each fragment is aligned to the genome of each parent separately (i.e. the mm9 reference sequence with annotated SNPs) and then merged the two alignments while retaining all distinct multiple alignments (a union of the set of all mapped fragments each identified by their mapping coordinate and read identifier). For comparison purposes, this alignment strategy is also applied in the TopHat and MapSplice pipelines.

2.3.3 Results from Simulated Data

In Table 2.2, I first present the average precision, recall, F scores, and Area Under the Curve (AUC) when LinearSVM, DecisionTree, and RandomForests were used to build the classification models. All scores were measured by 10-fold cross-validation. The results demonstrate that our feature set is adequate and can help detect unexpressed genes efficiently. The RandomForests is the best and most consistent among all three methods. The classification model trained by RandomForests can detect near 90% spurious calls due to misalignments. Though SVM has a slightly higher precision score, the recall is much lower than RandomForests. This is because RandomForests is more suitable than SVM for data with discrete features and is more powerful in handling correlations between features. Therefore, RandomForests are chosen as the default classification method for our GeneScissors pipeline.

Statistics	LinearSVM	DecisionTree	RandomForests
Precision	81.9%	83.7%	**89.6%**
Recall	83.0%	84.8%	**87.8%**
F-measurement	85.7%	84.2%	**88.6%**
AUC	0.843	0.837	**0.910**

Table 2.2: Summary of the results from different classification methods

Next, in order to understand how much improvement GeneScissors could bring to the overall transcriptome calling by correcting fragment misalignment, the results of improved GeneScissors pipelines are compared with those from the TopHat and Map-

Splice's pipelines. Both GeneScissors pipelines used the modified version of Cufflinks. The GeneScissors (TopHat) pipeline used the modified version of TopHat. The MapSplice and TopHat pipelines used the regular version of Cufflinks. The following three measurements are used to compare the performance at the gene level:

$$GenePrecision = \frac{Number\ of\ Correct\ Genes}{Number\ of\ Reported\ Genes}, \tag{2.6}$$

$$GeneRecall = \frac{Number\ of\ Correct\ Genes}{Number\ of\ Simulated\ Genes}, \tag{2.7}$$

$$GeneF - measurement = 2 \times \frac{GenePrecision \times GeneRecall}{GenePrecision + GeneRecall}. \tag{2.8}$$

The results of different pipelines are summarized in Table 2.3. All statistics are averaged over a 10-fold cross validation. I observe that Cufflinks tends to report a much higher number of genes in all four pipelines. There are only approximately 13000 expressed genes but Cufflinks reports more than 30000 genes in the TopHat or MapSplice pipelines and over 26000 genes in the GeneScissors pipelines.

A significant percentage of these reported genes can be mapped back to the "expressed" genes from which synthetic reads are generated. In fact, several reported genes are often mapped back to the same expressed gene by Cuffcompare. Cufflinks failed to recognize them as (possibly different transcripts of) the same gene, perhaps due to both the length and variable number of splice junctions and/or the low fragment coverage seen for some transcripts. In this case, when computing GenePrecision and GeneRecall, only one of them was counted as the "correct" gene, the remaining ones were counted as "incorrect" genes. Since all four pipelines used Cufflinks to infer transcriptome, all of them had relatively low GenePrecision. The GeneScissors (MapSplice) pipeline had a 12.6% improvement in GenePrecision over the original MapSplice pipeline, at the cost of a slight drop in GeneRecall. The GeneScissors (TopHat) pipeline had a 6.5% improvement in GenePrecision over the TopHat pipeline, while retaining the same level of GeneRecall. GeneScissors was able to detect and remove more than 4000 spurious (gene) calls by correcting fragment misalignments.

MapSplice pipeline has the highest score on GeneRecall, but a much lower GenePrecision score comparing with TopHat pipeline and GeneScissors pipeline. This is because MapSplice can find more possible alignments than TopHat, but is not able to identify the correct alignment when a fragment has multiple alignments. Hence, the MapSplice pipeline reported more false positives than the TopHat pipeline.

Statistics	MapSplice Pipeline	TopHat Pipeline	GeneScissors (MapSplice)	GeneScissors (TopHat)
Number of Reported Genes	36516	30622	26556	26473
GenePrecision	35.6%	41.8%	48.2%	**48.3%**
GeneRecall	**95.1%**	93.2%	93.0%	93.2%
GeneF-measurement	51.5%	58.2%	63.5%	**63.6%**

Table 2.3: Comparison of MapSplice, TopHat, GeneScissors (MapSplice) and GeneScissors (TopHat) pipelines.

Overall, the GeneScissors (TopHat) pipeline performed best among the four pipelines on this challenging test case. It is obvious that (1) detecting and correcting fragment misalignments can improve the accuracy in transcriptome inference under all circumstances; (2) given the correct fragment alignments, better transcriptome inference algorithms are still needed. In addition, GeneScissors does not assume all pseudogenes are unexpressed. In fact, GeneScissors is able to distinguish expressed pseudogenes from the rest with a comparable accuracy, demonstrated by a simulation study in the supplementary material.

2.3.4 Results from Real RNA-Seq Data

Both TopHat pipeline and our GeneScissors (TopHat) pipeline are also applied on the real RNA-Seq data. The running time for TopHat pipeline was about 24 hours per sample, and the extra running time for GeneScissors (TopHat) pipeline were approximately 10 hours per sample. Overall, the GeneScissors (TopHat) pipeline reported 4.25% fewer transcripts in real data than the TopHat pipeline (Figure 2.4 (a)). Considering that GeneScissors removed most of false positives in our simulation study, it suggests that the transcripts reported by the TopHat pipeline include a significant number of false positives.

Despite the fewer number of transcripts reported by GeneScissors, Figure 2.4(b) shows that GeneScissors actually reported 0.97% more transcripts that exactly match or partially match the splice junction annotations in the Ensembl database than the TopHat pipeline (The improvement is statistically significant with a p-value lower than 10^{-14} under the paired student's t-test). These transcripts are likely the false negatives missed by the TopHat pipeline due to misalignments. Figure 2.4(c) shows that the TopHat pipeline reported over 800 transcripts that are annotated as pseudogenes in Ensembl. GeneScissors managed to remove over 53.6% of them, and the fraction of transcripts that overlap any pseudogenes decreased from 3.2% to 1.57%. Figure 2.4(d) shows that GeneScissors reported 16% fewer unannotated transcripts than the TopHat pipeline. All these results indicate that GeneScissors is effective in detecting and correcting false positive and false negative transcript reports caused by fragment misalignments.

Furthermore, the number of pseudogenes reported by the original TopHat/Cufflinks pipeline in inbred samples is fewer than the number in F1 hybrids. Similarly, the fraction of pseudogenes ($\sim 57\%$) removed by GeneScissors in the inbred samples is smaller than the fraction ($\sim 36\%$) removed in the F1 hybrids. This indicates that the additional complications of F1 samples pose additional challenges to RNA-Seq analysis pipelines, and makes them more prone to errors than the inbred samples.

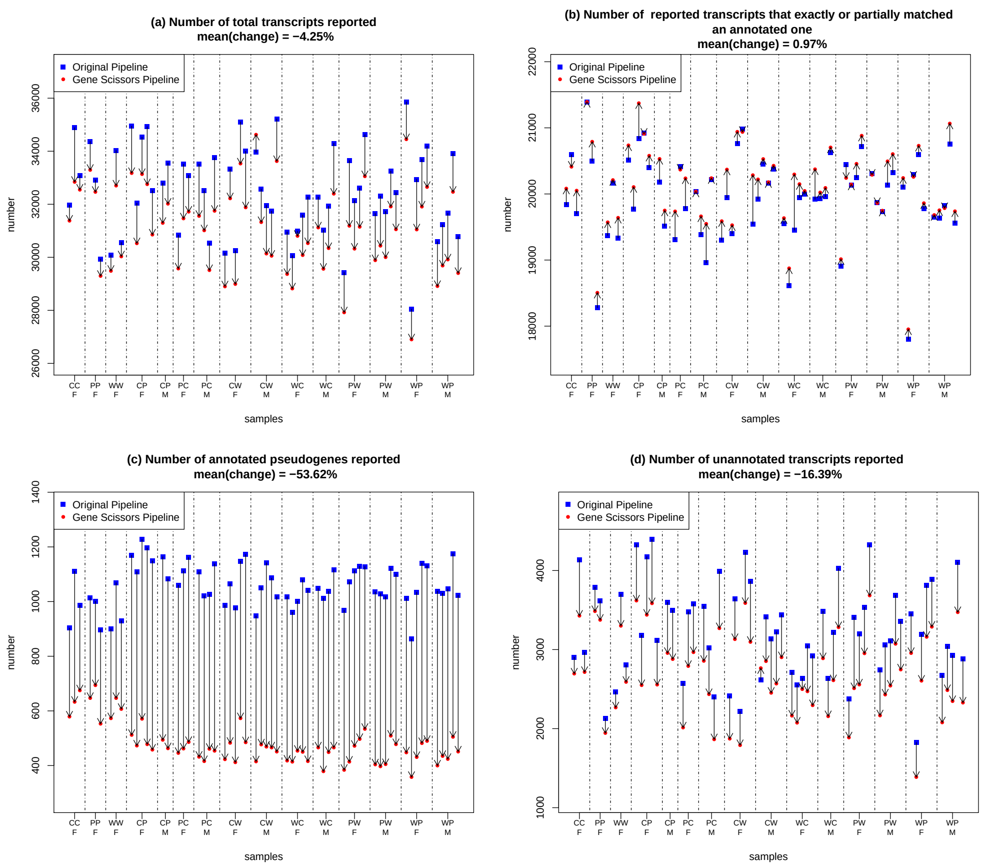


Figure 2.4: Comparisons between multiple samples run through both the GeneScissors pipeline and the TopHat pipeline. Results from the same sample are connected by an arrow. The three strains used were CAST/EiJ, PWK/PhJ, and WSB/EiJ, and they are indicated by the initials C, P, and W respectively. The two letter designations indicate the direction of the cross with the initial of the maternal strain followed by the initial of the paternal strain. The samples are clustered according to replicates from the same sex and F1 cross, followed by the reciprocal cross. The sex is indicated by F(female) and M(male).

2.4 Discussion and Conclusion

In this chapter, I present GeneScissors, a general approach to detect and correct transcriptome inference errors caused by misalignments, that can be applied to any RNA-Seq analysis pipeline. GeneScissors considers three underlying biological factors that lead to fragment misalignments and spurious transcript reporting. I propose a classification model to detect false discoveries due to misalignment, and the results show that it can provide significant improvement in overall accuracy.

Other heuristic approaches have been used to avoid reporting unexpressed genes in the RNA-Seq assembly result, such as discarding all known pseudogenes reported by the TopHat pipeline, masking repeated elements in genome, or aligning fragments to known transcriptome instead of genome. The key difference is that our RNA-Seq analysis does not require any additional annotations beyond adding SNPs, and it still supports a novel "transcript discovery".

Transcript discovery is important because current annotations are incomplete with regard to genes, isoforms, and allele specific variants. For example, in the real data, about 4000 unannotated transcripts clustered around 2300 unannotated genes on average are discovered. These transcripts persist after applying GeneScissors, which attempts to identify and correct misaligned fragments. This implies that current annotations are neither complete nor entirely accurate. For example, recent studies (Hirotsune et al., 2003; Khelifi et al., 2005) found that some regions previously thought to be pseudogenes can actually be transcribed to mRNA. Hence, removing all annotated pseudogenes, or highly repeated regions may lead to the removal of actual expressed transcripts. In contrast, GeneScissors might choose a pseudogene over the annotated paralog based on which better matches known genetic variants.

Furthermore, current pipelines using Cufflinks tend to overreport genes, especially when the genes share a high degree sequence similarity with other expressed genes in the data. The problem is alleviated to some extent by GeneScissors by recovering missed multiple fragment alignments and discarding fragment alignments to unexpressed genes/regions. However, there is still room for improvement.

CHAPTER 3

RNA-SKIM: A RAPID METHOD FOR RNA-SEQ QUANTIFICATION AT TRANSCRIPT-LEVEL

3.1 Introduction

In Chapter 2, I elaborate my approach to correct errors in the RNA-Seq alignment and assembly steps in the alignment-dependent pipelines to improve the accuracy of these methods. In this chapter, I introduce a novel alignment-free method, focusing on improving the efficiency and computational performance of RNA-Seq quantification.

Various aligners (TopHat (Trapnell et al., 2009), SpliceMap (Au et al., 2010), MapSplice (Wang et al., 2010)) are devised to infer the origin of each RNA-Seq fragment in the genome. The alignment step is usually time-consuming, requiring substantial computational resources and demanding hours to align even one individual's RNA-Seq data. Since there are multiple variations of RNA-Seq sequencing techniques, e.g. single-end sequencing and paired-end sequencing, to facilitate the discussion in this chapter, I simply refer to the *read* from single-end sequencing or the *pair of reads* from paired-end sequencing as a *fragment*. More importantly, a significant percentage of the fragments cannot be aligned without ambiguity, which yields a complicated problem in the quantification step: how to assign the ambiguous fragments to compatible transcripts and to accurately estimate the transcript abundances. Chapter 2 introduces a method to remove the errors of aligning a fragment to an unexpressed gene, but what if the fragments are assigned to multiple expressed genes?

To tackle the fragment multiple-assignment problem, an expectation-maximization

(EM) algorithm (Pachter, 2011) is often employed to probabilistically resolve the ambiguity of fragment assignments: at each iteration, it assigns fragments to their compatible transcripts with a probability proportional to the transcript abundances, and then updates the transcript abundances to be the total weights contributed from the assigned fragments, until a convergence is reached. The EM algorithm's simplicity in its formulation and implementation makes it a very popular choice in several RNA-Seq quantification methods (Cufflinks (Trapnell et al., 2010), Scripture (Guttman et al., 2010), RSEM (Li and Dewey, 2011), eXpress (Roberts and Pachter, 2013)). Since all fragments and transcripts are quantified at the same time in the EM algorithm, it usually requires considerable running time. Some studies (IsoEM (Nicolae et al., 2011), MMSEQ (Turro et al., 2011)) reduced the scale of the problem by collapsing reads if they can be aligned to the same set of transcripts. It is also worth mentioning that RNA-Seq quantification is an important first step for differential analysis on the transcript abundances among different samples (Trapnell et al., 2012b).

The alignment step is a vital step in the RNA-Seq assembly study (Trapnell et al., 2010) and has become the computational bottleneck for RNA-Seq quantification tasks. If users are only interested in RNA-Seq quantification of an annotated transcriptome, aligning RNA-Seq fragments to the genome seems cumbersome: not only do the RNA-Seq aligners take a long time to align fragments to the genome by exhaustively searching all possible splice junctions in the fragments, they may also generate misaligned results due to repetitive regions in the genome or sequencing errors, introducing errors in the quantification results (Zhang et al., 2013).

From another perspective, the annotation databases of transcriptome, e.g. RefSeq (Pruitt et al., 2007) and Ensembl (Flicek et al., 2011), play an increasingly important role in RNA-Seq quantification. For example, TopHat/Cufflinks supports a mode that allows users to specify the transcriptome by supplying an annotation database (a GTF file). RSEM (Li and Dewey, 2011) uses bowtie (Langmead et al., 2009) — a DNA sequence aligner — to align fragments directly to the transcriptome. Aligning RNA-Seq fragments to transcriptome avoids the computation to detect novel splice junctions in fragments and eliminates the non-transcriptome regions in the genome from further examination, and thus reduces the total running time of the quantification method and the number of erroneous alignments in the results.

To further improve the performance, the use of k-mers was recently proposed. The concept of k-mers — short and consecutive sequences containing k nucleic acids — has been widely used in bioinformatics, including genome and transcriptome assembly (Grabherr et al., 2011b; Fu et al., 2014), error correction in sequence reads (Le et al., 2013), etc. Since the number of k-mers in the genome or transcriptome is enormous, the need to store of all k-mers impedes the counting step of the k-mers. Most of existing methods save memory usage during the computation by using sophisticated algorithms and advanced data structures (bloom filter (Melsted and Pritchard, 2011), lock-free memory-efficient hash table (Marcais and Kingsford, 2011), suffix array (Kurtz et al., 2008)) or relying on disk space to compensate memory space (Rizk et al., 2013).

Thanks to the recent advances in both annotated transcriptome and algorithms to rapidly count k-mers, the transcriptome-based alignment-free method, Sailfish (Patro et al., 2013), requires 20 times less running time and generates comparable results to alignment-dependent quantification methods. Sailfish is a very lightweight method: it first builds a unique index of all k-mers that appear at least once in the transcriptome, counts the occurrences of the k-mers in the RNA-Seq fragments, and quantifies the transcripts by the number of occurrences of the k-mers through an EM algorithm. And surprisingly, the optimization problem underlying the EM algorithm developed in Sailfish is almost identical to that in RSEM and other methods, except that Sailfish assigns k-mers (instead of fragments) to transcripts.

Regardless of being alignment-dependent or alignment-free, all methods need to recover the fragment depth — the number of fragments that cover a specific location — across the whole transcriptome as one of the initial steps. However, none of the existing methods exploit the strong redundancy of the fragment depth in RNA-Seq data. More specifically, Fig. 3.1 shows a strong correlation between the fragment depth of any two locations that are a certain distance apart on the transcriptome, varying the distance from 1 base-pair to 100 base-pairs. Even when the two locations are 20 base-pair away from each other, the Pearson correlation score is still as high as 0.985. In other words, if an RNA-Seq quantification method that is able to recover the fragment depths for every 20 base-pairs and quantify the abundance levels based on such information, there should be no significant accuracy loss in the result. Recently, Uziela and Honkela (2013) developed a method that simply counts the number of alignments that covers the locations of hybridization probes used in the gene expression studies. Though these probes only represent a sparse sampling on every transcript in the transcriptome, the method still provides reasonably accurate results. These observations inspire us to ask the question: what is the minimum information we need in order to achieve comparable accuracy to state-of-the-art methods in RNA-Seq quantification? More specifically, does there exist a subset of k-mers that can provide accurate transcriptome quantification? And if so, how do we identify and use them to quantify transcriptome efficiently?

To answer these questions, I develop a similarity based clustering method to partition the transcriptome into separate clusters, and for each cluster of transcripts, I introduce a special type of k-mers called *sig-mers* which only appear in the cluster. Based on these sig-mers, RNA-Skim (Zhang and Wang, 2014) is developed, which is much faster than Sailfish and also maintains the same level of accuracy in the results. RNA-Skim includes two stages, *preparation* and *quantification*. In the preparation stage, RNA-Skim first partitions transcripts into clusters, and uses bloom filters to discover all sig-mers for each transcript cluster, from which a small yet informative subset of sig-mers are

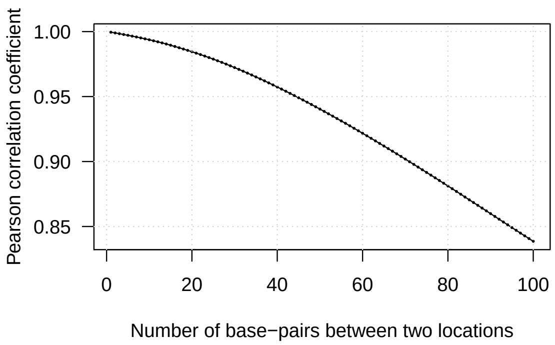


Figure 3.1: This figure shows the correlations of the fragment depth of any pair of locations as a function of the distance between the two locations from 1 base-pair to 100 base-pairs. This figure is generated based on the alignments reported by TopHat on a real RNA-Seq data.

selected to be used in the quantification stage. In the quantification stage, a rolling hash method (Karp and Rabin, 1987) is developed to rapidly count the occurrences of the selected sig-mers, and an EM algorithm is employed to properly estimate the transcript abundance levels using the sig-mer counts. Since no sig-mer is shared by two transcript clusters, the task can be easily divided into many small quantification problems, which significantly reduces the scale of each EM process and also makes it trivial to be parallelized. While RNA-Skim provides similar results to those of alternative methods, it only consumes 10% of the computational resources required by Sailfish.

In this chapter, I first describe the RNA-Skim method, then compare RNA-Skim with other methods, followed by the experimental results using both simulated and real data.

3.2 Method

In this section, I introduce the notion of sig-mer, which is a special type of k-mer that may serve as a *signatures* of a cluster of transcripts, distinguishing them from transcripts in other clusters in the transcriptome that do not contain this k-mer.

3.2.1 Sig-mer

In this paper, an annotated *transcriptome* Θ consists of a set of T transcripts: $\Theta = \{t_1, ..., t_T\}$. A *transcript* t represents an RNA sequence, essentially composed of a string of four nucleotide bases 'A', 'U', 'C', and 'G'. We use the corresponding four DNA nucleotides 'A', 'T', 'C', and 'G' to present the string of a transcript in the following. A *partition* of a given transcriptome Θ groups all transcripts into P disjoint non-empty subsets or clusters, denoted by $\Phi(\Theta) = \{\phi_1, ..., \phi_P\}$. For example, one commonly adopted partition of transcriptome is to group transcripts into genes based on their locations on the genome. For any transcript t, I use $\phi(t)$ to denote the cluster to which t belongs.

A substring of length k from a transcript sequence, its reverse sequence, its complimentary sequence, or its reverse and complimentary sequence is called *a k-mer of the transcript*. I define a function $\textbf{k-mer}()$ to represent the set of all k-mers from a single transcript or a cluster of transcripts, denoted as $\textbf{k-mer}(t)$ or $\textbf{k-mer}(\phi_p)$ respectively. For simplicity, if a string s is a k-mer of transcript t, I say $s \in \textbf{k-mer}(t)$. In this case, $s \in \textbf{k-mer}(\phi(t))$ is also true.

Definition 1 *Given a length k, a transcriptome Θ and its partition $\Phi(\Theta)$, if a k-mer s only exists in one cluster ϕ_p and never appears in other clusters $\Theta \backslash \phi_p$, it is named a* sig-mer *of cluster ϕ_p in this Chapter. And for any given cluster ϕ_p, all of its sig-mers is denoted as $\Omega(\phi_p)$. That is,*

$$\Omega(\phi_p) = \{s | s \in \textbf{k-mer}(\phi_p), \forall \phi_q \in \Theta \backslash \phi_p, s \notin \textbf{k-mer}(\phi_q)\}.$$

Sig-mers characterize the uniqueness of each cluster. It is obvious that the number of sig-mers heavily depends on the transcriptome partition. If transcripts with similar sequences are assigned to different clusters, k-mers from these transcripts may not qualify as sig-mers. Consequently, fewer sig-mers may be identified, and in the worst case, some cluster may not have any sig-mers.

3.2.2 Workflow of RNA-Skim

Since sig-mers are unique to only one cluster of transcripts, if a sig-mer occurs in some RNA-Seq reads, it indicates the sig-mer's corresponding transcripts may be expressed. Therefore, its occurrence in the RNA-Seq data may serve as an accurate and reliable indicator of the abundance levels of these transcripts. I propose a method, RNA-Skim, for quantifying the transcripts using the sig-mer counts in RNA-Seq data. Since no sig-mer is shared between transcript clusters, the problem reduces to quantifying transcript abundances using sig-mer counts within each cluster, which can be solved much more efficiently and can be easily parallelized. This is different from Sailfish that uses all k-mers in the transcriptome. In fact, RNA-Skim can be considered as a generalization of Sailfish: if the whole transcriptome is treated as a single cluster that includes all transcripts, all k-mers become sig-mers, and RNA-Skim degenerates to the exact formulation of Sailfish.

The workflow of RNA-Skim includes two stages: *preparation* and *quantification*. In preparation, RNA-Skim clusters the transcripts based on their sequence similarities, finds all sig-mers for each transcript cluster, and selects a subset of sig-mers to be used in the quantification stage. In quantification, RNA-Skim quickly counts the occurrences of sig-mers and quantifies transcripts within each cluster. The preparation stage of RNA-Skim does not require RNA-Seq read data and thus can be computed once as an offline process and be repeatedly used in the quantification stage.

3.2.3 Preparation Stage

In the preparation stage, RNA-Skim only requires users to supply a transcriptome (including all transcript sequences and gene annotations) and specify a desired sig-mer length to be used in RNA-Skim.

Transcript Partitioning A straightforward way to partition transcripts is to use the definitions of genes, which are based on their genome locations from an annotation database. However, the result of this partitioning approach may not be optimal because some transcripts of different genes may have similar sequences or share common subsequences. In order to minimize the number of common k-mers shared between clusters, RNA-Skim uses a sequence similarity based algorithm to generate a partition of transcriptome, instead of using any existing partition. I first define the k-mer-based similarity, which is used as the sequence similarity in the algorithm.

Definition 2 *The k-mer-based similarity of two sets of sequences ϕ_i and ϕ_j is defined as the higher of the two ratios: the number of common k-mers divided by the total number of k-mers in ϕ_i, and the number of common k-mers divided by the total number of k-mers in ϕ_j:*

$$\mathbf{k\text{-}mer\text{-}Similarity}(\phi_i, \phi_j) = \tag{3.1}$$
$$\max(\frac{|\mathbf{k\text{-}mer}(\phi_i) \cap \mathbf{k\text{-}mer}(\phi_j)|}{|\mathbf{k\text{-}mer}(\phi_i)|}, \frac{|\mathbf{k\text{-}mer}(\phi_i) \cap \mathbf{k\text{-}mer}(\phi_j)|}{|\mathbf{k\text{-}mer}(\phi_j)|}). \tag{3.2}$$

Transcripts from the same gene are very likely to be similar to each other. Hence, it is very likely that these transcripts will be assigned to the same cluster. To avoid unnecessary computation, RNA-Skim operates at the level of genes, rather than transcripts. However, calculating the exact similarity between a pair of genes requires generating all k-mers appearing in each gene and taking the intersection of the two sets. This is computationally expensive. To expedite the computation, RNA-Skim employs the data structure — bloom filter (Bloom, 1970) — coupled with a sampling based approach to approximate the similarity between two genes. The bloom filter is a space-efficient probabilistic data structure that is used to test whether an element is a member of a set, without the need of storing the set explicitly[1]. It can be maintained efficiently when new members are added to the set.

RNA-Skim first builds a bloom filter for the set of k-mers of each gene. Then, it randomly samples two subsets of k-mers — noted as $X(\phi_i)$ and $X(\phi_j)$ — from the pair of genes , and the $\mathbf{k\text{-}mer\text{-}Similarity}(\phi_i, \phi_j)$ is approximated[2] by $\max(\frac{|X(\phi_i) \cap \mathbf{k\text{-}mer}(\phi_j)|}{|X(\phi_i)|}, \frac{|\mathbf{k\text{-}mer}(\phi_i) \cap X}{|X(\phi_j)|}$ After I calculate the approximated similarities for every pair of genes, an undirected graph is built with each node representing a gene. There is an edge between two nodes if the similarity of the two corresponding genes exceeds a user-specified threshold γ. Each connected component of this graph naturally forms a cluster of nodes; each cluster of nodes forms a cluster of genes and transcripts of the genes.

Sig-mer discovery By definition, the sig-mers are essentially the k-mers occurring in only one cluster of transcripts. A brute force way to find all sig-mers is, for every k-mer in the transcriptome, to determine whether the k-mer that appears in one cluster also appears in some other cluster. Because the number of possible k-mers is in the order of billions, without any sophisticated data structure and data compression algorithms, storing the k-mers alone will easily take at least tens of gigabytes of memory space which is way beyond the capacity of any commodity computer.

RNA-Skim again utilizes bloom filters to reduce memory usage. Three types of bloom filters are employed: a bloom filter $BF.ALL$ for checking whether a given k-mer has been examined or not, a bloom filter $BF.DUP$ for checking whether a given k-mer appears in more than one cluster or not, and a bloom filter $BF.S(\phi_p)$ for each

[1] A bloom filter may yield a small number of false positives, but no false negatives. The false positive rate is bounded if the number of elements in the set is known.

[2] Our experiments show that we only need a small number (e.g., 10) of k-mers from each gene to achieve approximation with high accuracy.

cluster ϕ_p for checking whether a given k-mer is in $\textbf{k-mer}(\phi_p)$.

First, for each cluster ϕ_p, all distinct k-mers in it are enumerated: RNA-Skim enumerates all k-mers for every transcript in the cluster, and adds them to $BF.S(\phi_p)$; if a k-mer is already in $BF.S(\phi_p)$, it will be ignored. Second, every distinct k-mer in ϕ_p is added into $BF.ALL$, and if it is already in $BF.ALL$ (that is, it was added when RNA-Skim examined other clusters), it is added into $BF.DUP$. Therefore, if a k-mer occurs in multiple clusters, it is added in $BF.DUP$. Last, every k-mers of the transcriptome is enumerated again, and if the k-mer is not in $BF.DUP$, it is reported as a sig-mer, since it only occurs in one cluster.

Since bloom filters may have false positive reports, but never have false negatives, through this approach, some genuine sig-mer strings may be missed, but a non-sig-mer will never be labeled as a sig-mer. Figure 3.2 shows the pseudocode of the algorithm to find sig-mers.

Sig-mers selection RNA-Skim does not use all sig-mers because they are still numerous. Instead, RNA-Skim selects a subset of sig-mers for the quantification stage. I used a simple approach to select sig-mers from all sig-mers found by the previous step: for every transcript, sig-mers are evenly chosen based on the sig-mer locations such that any two sig-mers are at least 50 base-pair away from each other in the given transcript. Since some sig-mers may appear in multiple transcripts in the same cluster, for every selected sig-mer, all transcripts are re-examined, and the ones that contain the sig-mer are also recorded. This approach guarantees that every transcript is associated with some sig-mers (as long as there exist some sig-mers). A good sig-mer coverage is crucial for accurate quantification of transcript abundance. The final output of the preparation step includes the partition of the transcriptome, selected sig-mers, and their associating clusters and transcripts.

```
foreach partition of transcripts $\phi_p \in \Theta$
    foreach location $l \in \phi_p$
        generate the k-mer $s$ at the location $l$
        if $s \notin BF.S(\phi_p)$ then
            Add $s$ into $BF.S(\phi_p)$
            if $s \notin BF.ALL$ then
                Add $s$ into $BF.ALL$
            else
                Add $s$ into $BF.DUP$
    end foreach
end foreach
foreach partition of transcripts $\phi_p \in \Theta$
    foreach location $l \in \phi_p$
        generate the k-mer $s$ at the location $l$
        if $s \notin BF.DUP$ then
            Report $s$ as a sig-mer of $\phi_p$.
    end foreach
end foreach
```

Figure 3.2: The pseudocode to find all sig-mers.

3.2.4 Quantification Stage

The quantification stage requires users to provide RNA-Seq data (e.g. FASTA files) and the selected sig-mers associated with transcripts containing them from the preparation stage.

Sig-mer counting Since the number of sig-mers used in RNA-Skim is much smaller than the number of k-mers typically used by other k-mer-based approaches, all sig-mers can be stored in a hash table in memory. The number of occurrences of all sig-mers can be counted by enumerating all k-mers in the RNA-Seq reads and looking up the k-mers in the hash table to update the corresponding counters. RNA-Skim basically follows this scheme with a tweak on the hash function to further speed up the computation.

In a straightforward implementation of the previously described algorithm, every k-mer incurs an O(k) operation to calculate its hash value, and this hash operation can be further reduced to O(1) by the Robin-Karp pattern matching algorithm (Karp and Rabin, 1987). The Robin-Karp pattern matching algorithm requires a special hash function — rolling hash — that only uses multiplications, additions and subtractions.

In rolling hash, the hash value $\mathbf{H}(r)$ of the first k-mer in the RNA-Seq read r is calculated by

$$\mathbf{H}(r[0,...,k-1]) = \boldsymbol{\chi}(r[0]) \times h^{k-1} + \boldsymbol{\chi}(r[1]) \times h^{k-2} + ... + \boldsymbol{\chi}(r[k-1]) \times h^{0},$$

where h is the base of the hash function, $r[i]$ is the ith character in s, and the character hash function $\boldsymbol{\chi}()$ maps a character to an integer value. One way to calculate the hash value for the (sequentially ordered) second k-mer $r[1,...,k]$ is

$$\mathbf{H}(r[1,...,k]) = \boldsymbol{\chi}(r[1]) \times h^{k-1} + \boldsymbol{\chi}(r[2]) \times h^{k-2} + ... + \boldsymbol{\chi}(r[k]) \times h^{0}.$$

But thanks to the structure of the rolling hash function, $\mathbf{H}(r[1,...,k])$ can be calculated in a much faster way:

$$\mathbf{H}(r[1,...,k]) = (\mathbf{H}(r[0,...,k-1]) - \boldsymbol{\chi}(r[0]) \times h^{k-1}) \times h + \boldsymbol{\chi}(r[k]) \times h^{0},$$

which only requires one subtraction, three multiplications and one addition. And I can continue to calculate the hash values for the subsequent k-mers in the fragment in this fashion. Thus, after the hash value of a given k-mer is calculated, the hash value can be looked up in the hash table, and if it is in the hash table, its associated counter is incremented accordingly. Since RNA-Skim uses this specially designed hash function, I implemented the hashtable in RNA-Skim using open addressing with linear probing. The base h is arbitrarily set to be a prime number 37, and the function $\boldsymbol{\chi}()$ maps every character to its actual ASCII value.

Quantification Since every cluster of transcripts has a unique set of sig-mers, which are the k-mers that never appear in other transcript clusters, every cluster can be independently quantified by RNA-Skim, resulting in a set of smaller independent quantification problems, instead of one huge whole transcriptome quantification problem in other approaches.

Formally, if ϕ_p is a cluster of transcripts, the set of sig-mers of ϕ_P is denoted by $S(\phi_p)$, a sig-mer is denoted by s , and the number of sig-mers that are contained by transcript t is denoted by b_t. From the previous steps, the following information can be obtained: c_s (the number of occurrences of the sig-mer s in the RNA-Seq data) and $y_{s,t}$ (binary variables indicating whether the sig-mer s is contained in the sequence of transcript t or not). C is the number of occurrences of all sig-mers ($C = \sum_s c_s$). An occurrence of a sig-mer in the RNA-Seq dataset is denoted by o ($o \in O(\phi_p)$) and its sig-mer is denoted by z_o, and the set of all occurrences of sig-mers is denoted by $O(\phi_p)$ (e.g., $s \in s(\phi_p)$).

Same as in the previous study (Pachter, 2011), we define $\boldsymbol{\Psi} = \{\alpha_t\}_{t \in \phi_p}$ where α_t is the proportion of all selected sig-mers that are included by the reads from transcript t, and $\sum \alpha_t = 1$. For an occurrence o, $p(o \in t)$ represents the probability that o is chosen from transcript t, in a generative model,

$$p(o \in t) = \quad y_{z_o,t}\tfrac{\alpha_t}{b_t} \tag{3.3}$$

Therefore, the likelihood of observing all occurrences of the sig-mers as a function of the parameter $\mathbf{\Psi}$ is

$$\begin{aligned}\mathcal{L}(\mathbf{\Psi}) &= \prod_{o \in O(\phi_p)} \sum_{t \in \phi_p} p(o \in t) & (3.4)\\ &= \prod_{o \in O(\phi_p)} \sum_{t \in \phi_p} y_{z_o,t} \frac{\alpha_t}{b_t} & (3.5)\\ &= \prod_{s \in S(\phi_p)} (\sum_{t \in \phi_p} y_{s,t} \frac{\alpha_t}{b_t})^{c_s}. & (3.6)\end{aligned}$$

This is in spirit similar to the likelihood function used in other studies, except that this is the likelihood of observing sig-mers rather than fragments (Li and Dewey, 2011) or k-mers (Patro et al., 2013). Thus, an EM algorithm is used to find $\mathbf{\Psi}$ that maximizes the likelihood: it alternates between allocating the fraction of counts of observed sig-mers to transcripts according to the proportions $\mathbf{\Psi}$ and updating $\mathbf{\Psi}$ given the amount of sig-mers assigned to transcripts.

Specifically, $\beta_{s,t}$ is the expected number of occurrences of sig-mer s assigned to transcript t, and in the expectation step of the EM algorithm, its value is computed by

$$\beta_{s,t} = c_s \frac{y_{s,t} \frac{\alpha_t}{b_t}}{\sum_{q \in \phi_p} y_{s,q} \frac{\alpha_q}{b_q}}. \tag{3.7}$$

In the maximization step, α_t can be estimated by $\frac{\sum_{s \in S(\phi_P)} \beta_{s,t}}{C}$, which is the ratio of the number of occurrences of sig-mers assigned to transcript t to the total number of occurrences of all sig-mers. The details of how these steps are derived can be found in Xing et al. (2006) RNA-Skim also applies the same technique used in Patro et al. (2013), Nicolae et al. (2011), and Turro et al. (2011) to collapse sig-mers if they are contained by the same set of transcripts.

In RNA-Seq, if the read length is R, there are R distinct reads that may cover a given position in a transcript, and $R - k + 1$ distinct reads to entirely cover a k-mer

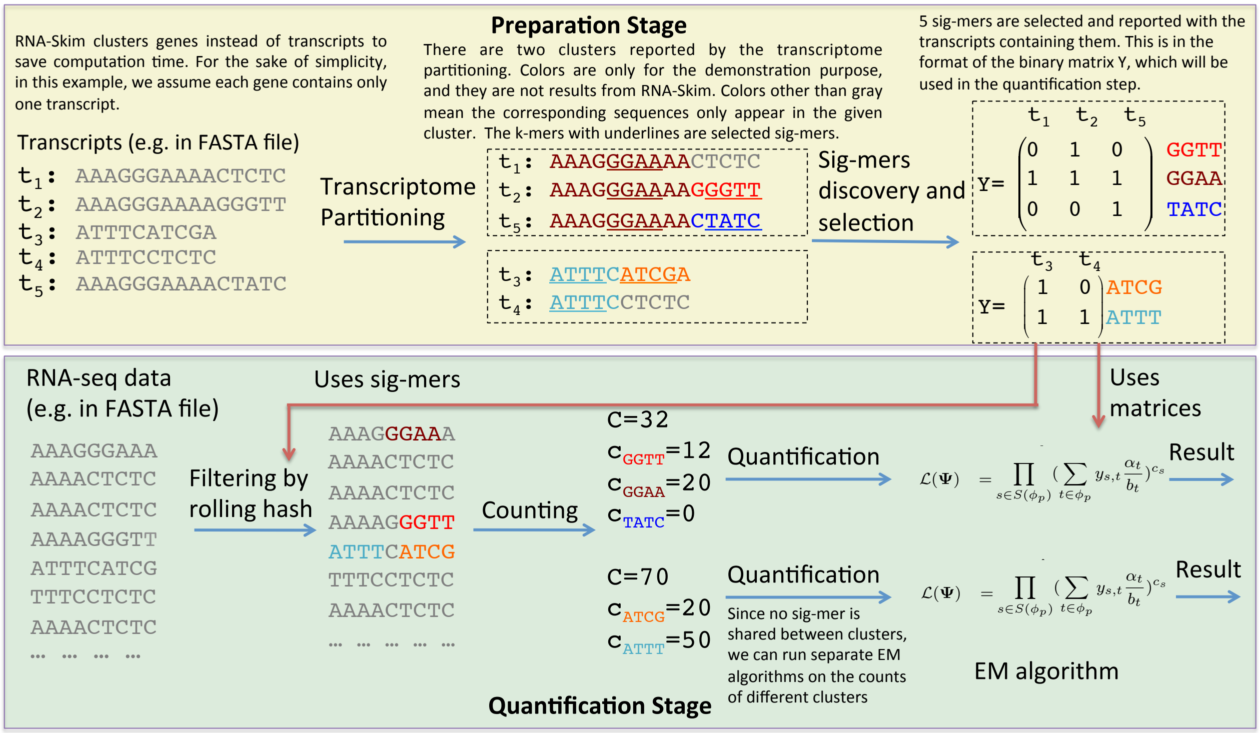

Preparation Stage
RNA-Skim clusters genes instead of transcripts to save computation time. For the sake of simplicity, in this example, we assume each gene contains only one transcript.
There are two clusters reported by the transcriptome partitioning. Colors are only for the demonstration purpose, and they are not results from RNA-Skim. Colors other than gray mean the corresponding sequences only appear in the given cluster. The k-mers with underlines are selected sig-mers.
5 sig-mers are selected and reported with the transcripts containing them. This is in the format of the binary matrix Y, which will be used in the quantification step.
Transcripts (e.g. in FASTA file)
t1: AAAGGGAAAACTCTC
t2: AAAGGGAAAAGGGTT
t3: ATTTCATCGA
t4: ATTTCCTCTC
t5: AAAGGGAAAACTATC
Transcriptome Partitioning
t1: AAAGGGAAAACTCTC
t2: AAAGGGAAAAGGGTT
t5: AAAGGGAAAACTATC
t3: ATTTCATCGA
t4: ATTTCCTCTC
Sig-mers discovery and selection
t1 t2 t5
Y=
0 1 0 GGTT
1 1 1 GGAA
0 0 1 TATC
t3 t4
Y=
1 0 ATCG
1 1 ATTT
RNA-seq data (e.g. in FASTA file)
AAAGGGAAA
AAAACTCTC
AAAACTCTC
AAAAGGGTT
ATTTCATCG
TTTCCTCTC
AAAACTCTC
... ...
Uses sig-mers
Filtering by rolling hash
AAAGGGAAA
AAAACTCTC
AAAACTCTC
AAAAGGGTT
ATTTCATCG
TTTCCTCTC
AAAACTCTC
... ...
Counting
C=32
c_GGTT=12
c_GGAA=20
c_TATC=0
C=70
c_ATCG=20
c_ATTT=50
Quantification
Quantification
Since no sig-mer is shared between clusters, we can run separate EM algorithms on the counts of different clusters
Uses matrices
$\mathcal{L}(\Psi) = \prod_{s\in S(\phi_p)} (\sum_{t\in\phi_p} y_{s,t}\frac{\alpha_t}{b_t})^{c_s}$
$\mathcal{L}(\Psi) = \prod_{s\in S(\phi_p)} (\sum_{t\in\phi_p} y_{s,t}\frac{\alpha_t}{b_t})^{c_s}$
Result
Result
EM algorithm
Quantification Stage

Figure 3.3: An illustration of how RNA-Skim works on a toy transcriptome of five transcripts.

starting at the given location. So, assuming the reads are uniformly sampled, $\frac{R\times\beta_{s,t}}{R-k+1}$ is the estimated abundance based on sig-mer s. When the EM algorithm converges, the transcript abundance μ_t can be calculated by averaging the abundance levels estimated by each sig-mer in the transcript,

$$\mu_t = \frac{\sum_{s\in\phi_p} R\times\beta s,t}{(R-k+1)\times b_t}, \tag{3.8}$$

RNA-Skim reports both Reads Per Kilobase per Million mapped reads (RPKM) and Transcripts Per Million (TPM) as the relative abundance estimations of the transcripts, and both metrics are calculated by the way used in Sailfish (Patro et al., 2013).

So far, I have described both preparation and quantification stages in RNA-Skim. In the last, a toy example is provided to illustrate how each stage works in RNA-Skim in Fig 3.3.

3.3 Software for comparison

RNA-Skim is implemented in C++ with heavy usage of the open-source libraries bloomd (Dadgar, 2013), protobuf (Google, 2013) and an open-source class StringPiece (Hsieh, 2013). The parameter settings will be discussed in the Results section.

I compared RNA-Skim with four different quantification methods: Sailfish (0.6.2), Cufflinks (2.1.1), RSEM (1.2.8), and eXpress (1.5.1) in both simulated and real datasets. TopHat (2.0.10) and Bowtie (1.0.0) are used as the aligners when needed.

For Sailfish, the k-mer size is set to be 31 because this value gives the highest accuracy in the simulation study, among all k-mer sizes supported by Sailfish ($k \leq 31$). For other software, I followed the experiments in Patro et al. (2013) to set the parameters. Input to Cufflinks was generated by TopHat which used Bowtie (–bowtie1) allowing up to three mismatches per read (-N 3 and –read-edit-dist 3). Both TopHat and Cufflinks were provided with a reference transcriptome. RSEM and eXpress directly used Bowtie to align the reads to the transcriptome, with the argument (-N 3) to allow up to three mismatches per read. The eXpress was executed in the streaming mode, to save the total quantification running time. For simulation study, estimations without bias correction for Sailfish, Cufflinks and eXpress are used for comparison. For real datasets, the estimations with bias correction are used for these three methods. For RSEM, since it does not provide an option to control the bias correction, we did not differentiate its usage in the simulation and real data studies. Other parameters were set to default values for these methods.

All methods were run on a shared cluster managed by the LSF (Load Sharing Facility) system. The running time and CPU time of these methods are measured by LSF. Each cluster node is equipped with Intel(R) Xeon(R) 12-core 2.93 GHz CPU and at least 48 GB memory. All files were served by the Lustre file system.

3.4 Materials

All materials including both simulated and real data are based on the mouse population and consist of paired-end reads with 100 base-pairs length per read. We used C57BL/6J downloaded from Ensembl (Build 70) as the reference genome in all experiments. All methods studied in this chapter were provided with 74215 protein-coding annotated transcripts from the Ensembl database. The simulation datasets, including 100 mouse samples with the number of reads varying from 20 millions to 100 millions, were generated by the flux-simulator (Griebel et al., 2012) with its default error model enabled. For real datasets, we used the RNA-Seq data from 18 inbred samples and 58 F1 samples derived from three inbred mouse strains CAST/EiJ, PWK/PhJ, and WSB/EiJ. The RNA-Seq data was sequenced from mRNA extracted from brain tissues of both sexes and from all 6 possible crosses (including the reciprocal).

3.5 Results

In this section, we first compared alternative partition algorithms and how they impact sig-mer selections in RNA-Skim and then furnish a comparison with four alternative methods on both simulated and real data. At last, we demonstrated that RNA-Skim is the fastest method among all considered methods.

3.5.1 Similarity-based Partition Algorithm

I compared the result of the similarity-based partition algorithm with those from two alternative ways to partition transcripts: transcript-based partition (every cluster contains a transcript) and gene-based partition (every cluster contains the transcripts from an annotated gene). The similarity threshold γ in the partition algorithm was set to be 0.2 (more details are provided later on the parameter choice). Table 3.1 compares these partitions on the same transcriptome. The number of clusters generated by the similarity-based partition is 20% fewer than the number of genes. The average number of transcripts per cluster is about 20% more than the average number of transcripts per gene. Most clusters only contain transcripts from a single gene, though the largest cluster contains 6107 transcripts. These transcripts in the largest cluster share a substantial number of k-mers (e.g., from paralogous genes) which need to be examined altogether in order to accurately estimate their abundance levels. Failing to consider them together (e.g., by using transcript-based or gene-based partitions) will compromise the number of sig-mers that help distinguish transcripts and hence impair the accuracy of transcriptome quantification. Even though this cluster contains many transcripts, it represents less than 10% of the total number of transcripts, which means the number of transcripts that RNA-Skim needs to quantify is 10% smaller than the Sailfish does.

type	number of clusters	average number of transcripts per cluster	size of the largest cluster
transcript	74215	1	1
gene	22584	3.29	39
RNA-Skim	18269	4.06	6107
Sailfish	1	74215	74215

Table 3.1: This table compares three different partitions. If the partition contains only one cluster of all transcripts, RNA-Skim degenerates to Sailfish. I thus listed it in the table for comparison.

these three types of partitions are used as the input to the sig-mer discovery method. I define the proportion of k-mers in a transcript that are sig-mers as the *sig-mer coverage* of the transcript. To evaluate the goodness of a partition, I measured t and plot the cumulative distribution of all transcripts sorted in ascending order of their sig-mer coverage in Fig. 6, with varying k-mer sizes. For any transcript, the higher the sig-mer coverage is, the more accurate the abundance estimation will be. The similarity-based partition is the best: almost all transcripts have at least 80% sig-mer coverage, which pushes the cumulative distribution curves to the lower right corner of the plot regardless of the k-mer size. The gene-based partition is slightly worse: about 95% of transcripts have at least 80% sig-mer coverage. The gene-based partition tends to result in low sig-mer coverage for genes sharing similar sequences. The transcript-based partition is the worst for an obvious reason: transcripts from the same genes may share exons and thus the number of sig-mers that can distinguish a transcript may be very small. I also observed that using longer k-mer improves the sig-mer coverage.

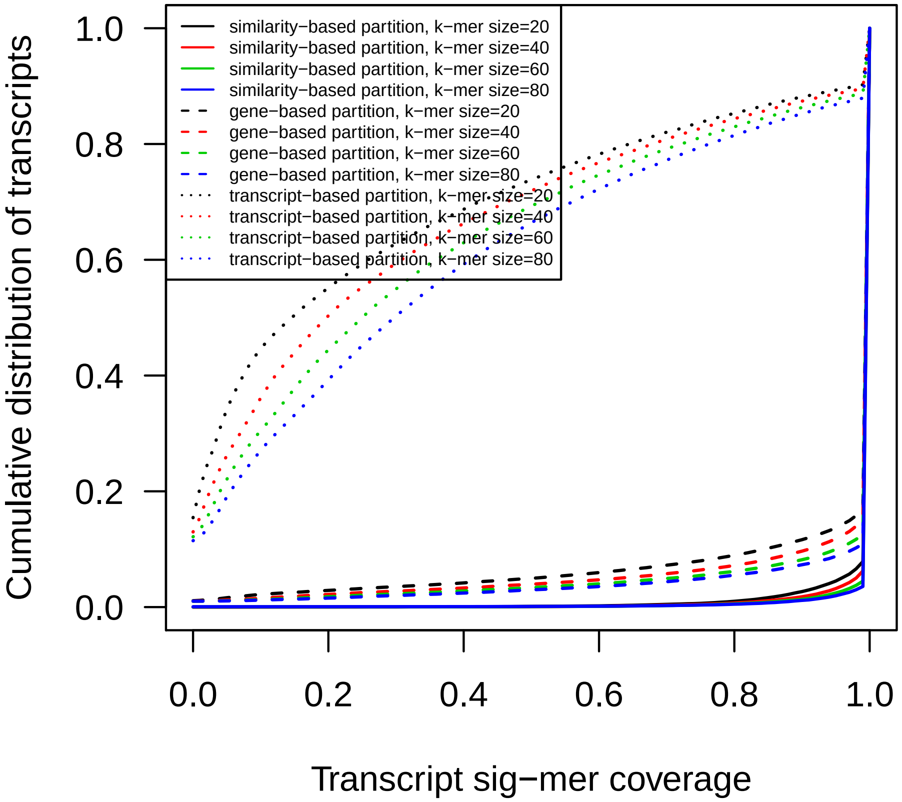
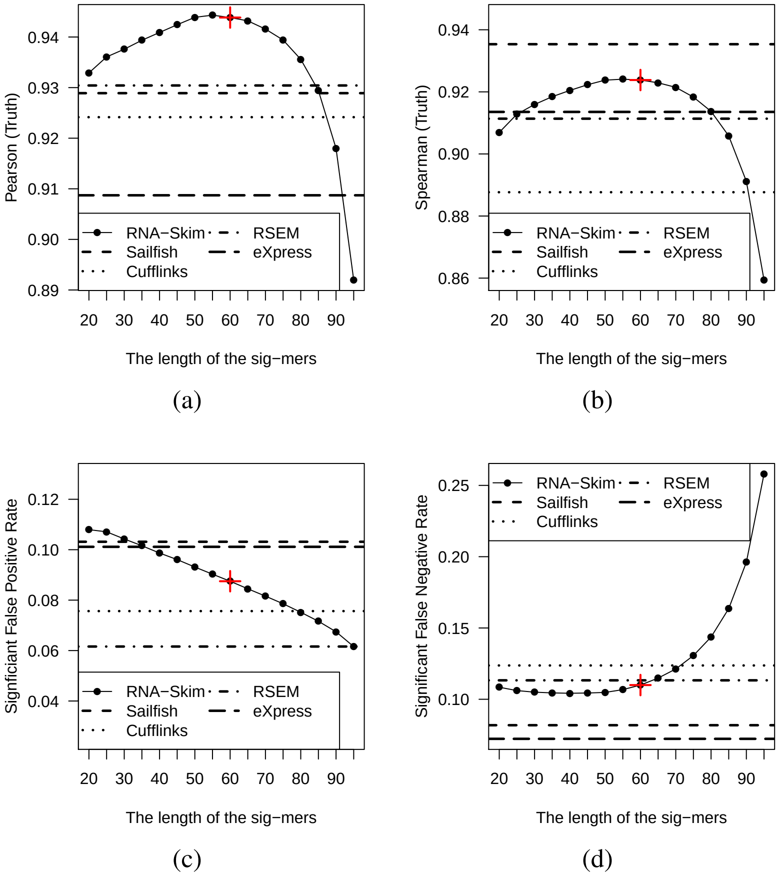


Figure 3.4: The cumulative distribution of all transcripts of their sig-mer coverage. The lower the curve is, the better the corresponding partition is.

In the end, RNA-Skim selects a total of 2586388 sig-mers to be used in the quantification stage, and these sig-mers count for less than 3.5% of 74651849 distinguished k-mers used by Sailfish. Since RNA-Skim uses a much smaller set of sig-mers, it is able to use the rolling hash method — a very fast but memory-inefficient method — to count sig-mers in RNA-Seq reads.

3.5.2 Simulation Study



(a) (b)

(c) (d)

Figure 3.5: These figures plot Pearson (Truth), Spearman (Truth), SFPR, and SFNR of RNA-Skim as a function of sig-mer length. For comparison, I also plotted that of the other four methods as the horizontal lines. The reported values are the average across 100 simulated samples. The red crosses indicate the sig-mer length (i.e., 60 base pairs) used in other experiments in this chapter.

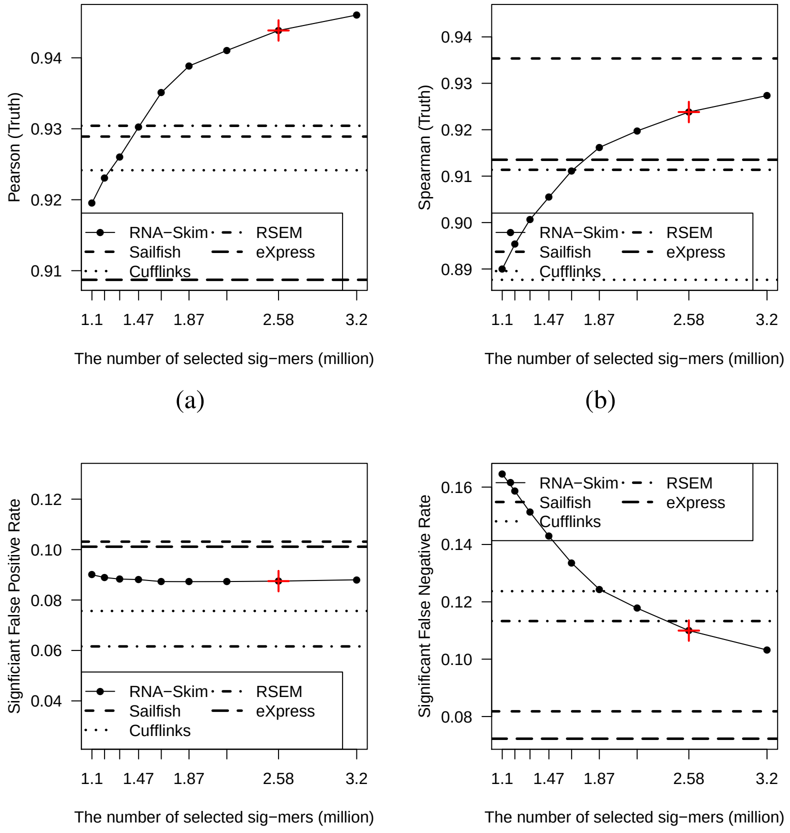
Fig. 3.5 compares the performance of the five methods on the simulated data using four metrics: Pearson's correlation coefficient, Spearman's rank correlation coefficient, significant false positive rate and significant false negative rate. For brevity, Pearson (Truth), Spearman (Truth), SFPR and SFNR are employed to denote these metrics respectively. The Pearson's correlation coefficient is calculated in a logarithmic scale, using all transcripts whose true and estimated abundance values are larger than 0.01 RPKM. This calculation is the same as that used by Sailfish (Patro et al., 2013). The

Spearman's rank correlation is calculated on the set of transcripts whose true abundance values are larger than 0.01 RPKM. If a transcript's estimation is larger than 0.1 RPKM, but its true abundance value is less than 0.01 RPKM (a 10 fold suppression), it is called as a *significant false positive*; similarity, if a transcript's estimation is smaller than 0.01 RPKM, but its true abundance value is larger than 0.1 RPKM (a 10 fold amplification), it is called as a *significant false negative*. The significant false positive rate and significant false negative rate are calculated to assess the estimation distributions on the set of transcripts excluded by the previous metrics. There are two reasons that I chose SFPR and SFNR instead of the regular false positive rate and false negative rate: first, we prefer the transcripts with relatively large abundance values (larger than 0.1 RPKM) because they are accountable for 99% the RNA-Seq data; second, due to the noisy nature of RNA-Seq, for the transcripts with small abundance values (less than 0.01 RPKM), it is very difficult to calculate accurately, e.g., both RSEM and Sailfish set the default minimal abundance value to be 0.01 RPKM.

For RNA-Skim, the sig-mer length are varied from 20 to 95 base-pairs. Other methods are presented as horizontal lines for comparisons. Despite the small differences by individual metrics, Fig. 3.5 shows that these five methods exhibit comparable performance: no method outperforms the remaining methods across all metrics and the maximal difference by any metric is within 0.05.

Fig. 3.5(a) and Fig. 3.5(b) show two concave curves of Pearson (Truth) and Spearman (Truth) for RNA-Skim by varying its sig-mer length. There are two factors explaining the concave curves. First, when the sig-mer length increases, sig-mers become more distinct, and the sig-mer coverage increases, which improves the correlations between the truth and estimation. Second, for any fixed read length, when increasing the sig-mer length, the probability that a sig-mer is contained by a single read drops, causing the decrease in the number of sig-mers observed in the RNA-Seq data, which may exacerbate the correlations. In summary, there is a clear trade-off on the sig-mer length. Empirically, the best sig-mer length is between 55 to 60, and I thus used 60 in other experiments.

For the same reason, in Fig. 3.5(c) and Fig. 3.5(d), I found that the increase in the sig-mer length affects positively on SFPR, but negatively on SFNR. Other methods also follow the same inverse correlation: while Sailfish and eXpress are the worst in SFPR among these five methods, they are the best two in SFNR.



(a)

(b)

(c)

(d)

Figure 3.6: These figures plot Pearson (Truth), Spearman (Truth), SFPR, and SFNR as a function of the number of sig-mers used in RNA-Skim. For comparison, I also showed that of the other four methods as horizontal lines. The reported values are the average across 100 simulated samples. The red crosses indicate the number of sig-mers (i.e., 2.58 million sig-mers) used in other experiments in this chapter.

Fig. 3.6 shows the Pearson (Truth), Spearman (Truth), SFPR, and SFNR as a function of the number of sig-mers used in RNA-Skim. In Fig. 3.6 (a)(b)(d), when the number of sig-mers increases, the three metrics improve substantially, though at different paces. Fig. 3.6 (c) shows no significant change in SFPR for different numbers of sig-mers. This observation suggests that we should use as many sig-mers as possible given available memory space. To ensure RNA-Skim to have similar memory usage to that of other methods, RNA-Skim uses 2.58 million sig-mers. This is also the default setting in other experiments in this chapter.

Table 3.2 shows that the metrics do not vary much when using different similarity thresholds. In the simulation study, the similarity threshold γ is varied from 0.06 to 0.28 and observed at most 0.005 change across all metrics. Hence, the detailed results for the thresholds between 0.06 (excluded) and 0.28 (excluded) are omitted.

γ	Pearson	Spearman	SFP	SFN
0.06	0.9438	0.9242	0.0692	0.0233
0.28	0.9440	0.9237	0.0698	0.0235

Table 3.2: This table shows that the four metrics do not change much for different similarity threshold γ.

Fig. 3.7 shows a strong and clear linear correlation between the estimated RPKM scores by RNA-Skim and the true RPKM scores on one simulated sample.

In simulation study, the accuracy of RNA-Skim depends on the sig-mer length and the number of sig-mers, but is insensitive to the threshold γ. When these parameters are chosen properly, RNA-Skim produces similar results to those by other methods.

3.5.3 Study using Real RNA-Seq data

Since the flux simulator cannot simulate RNA-Seq data with bias effects, and there might also be other unknown factors in the real RNA-Seq data that the simulator fails to capture, RNA-Skim is also compared with other methods on real data. Since the

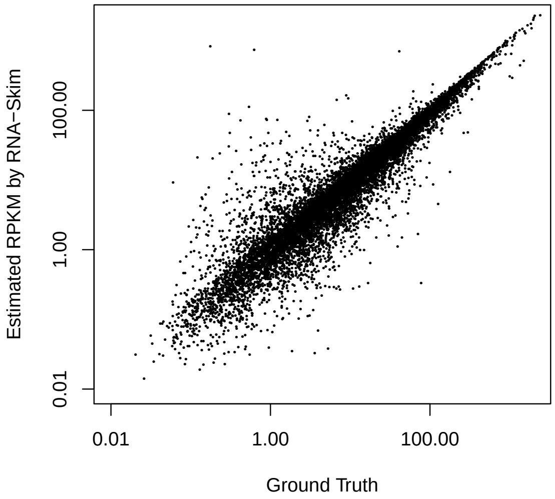


Figure 3.7: The scatter plot of the estimated RPKM scores by RNA-Skim vesus the true RPKM scores. Both axes are in a logarithmic scale, and all transcripts whose true RPKM or estimated RPKM is less than 0.01 are omitted.

ground truth on real data is not known, the Pearson correlation and Spearman correlation is computed between the results produced by RNA-Skim and one other method, referred to as Pearson (methods) and Spearman (methods) to distinguish from the previous computed correlations between RNA-Skim result and the ground truth.

Fig. 3.8 shows that the distributions of the Pearson (methods) and Spearman (methods) are not significantly different between real data and simulated data. For example, the differences between the mean values of the correlations on both simulated and real data are no more than 0.02 across all methods. This consistency suggests that the result from RNA-Skim may have similar correlations with the unobserved truth. The slightly wider distribution of the correlations in real data (than that in simulated data) suggests the real data may exhibit more diversity than simulated data.

3.5.4 Running Time

For the preparation stage (including transcriptome partitioning and sig-mer selection), RNA-Skim takes about 3 hours to finish on the mouse transcriptome by using a

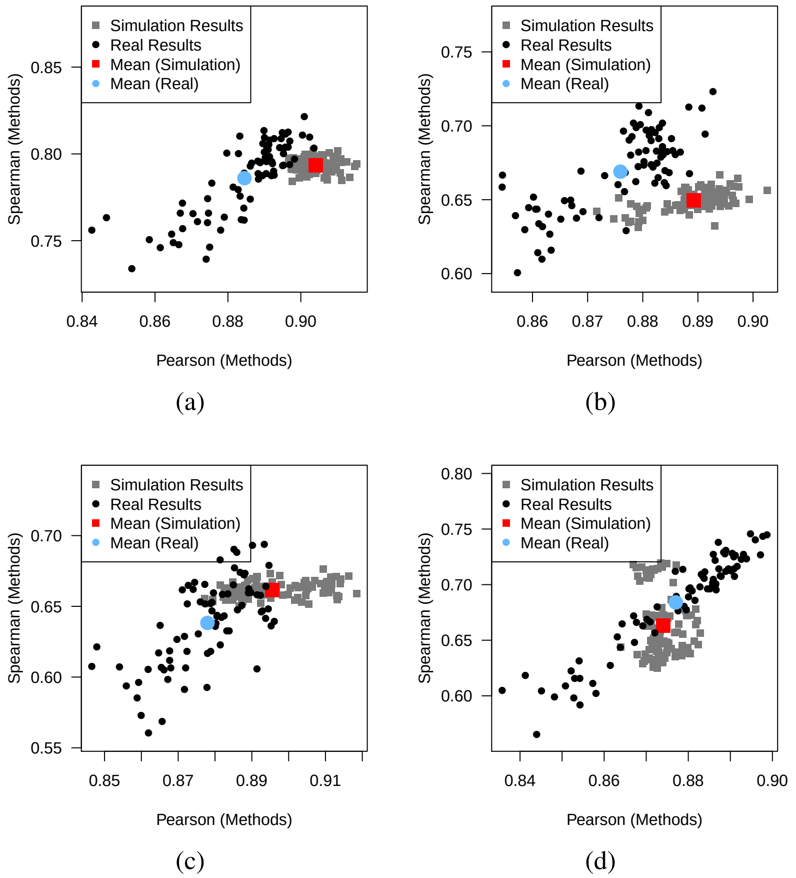


Figure 3.8: The distributions of the Pearson (methods) and Spearman (methods) correlations between the results from RNA-Skim and the results from each of the remaining methods on both simulated and real data.

single thread. Most time is spent on calculating the k-mer-based similarities between different pairs of genes. It takes about 10 minutes to finish sig-mer discovery and selection. It is worth noting that these steps only need to be run once for one population beforehand, and after sig-mers are selected and their connections with transcripts are established, the result can be repeatedly used on quantifying the transcriptome of many samples. Therefore, the running time for the preparation stage is less critical than the running time of the quantification stage, and the one-time investment of 3 hours is acceptable.

For the quantification stage, I compared both the running time and the CPU time of

Method	Number of threads	Running time (sec)	CPU time (sec)	Speedup (CPU time)
RNA-Skim	1	592	592	1x
Sailfish	8	972	7005	11.8x
TopHat + Cufflinks	8	12480	68834	116x
Bowtie + RSEM	8	17160	79222	133x
Bowtie + eXpress	8	13800	111273	188x

Table 3.3: This table shows the running time of these five methods on a real sample with 44 millions of paired-end reads.

these five methods on a real sample with 44 millions of paired-end reads. The running time is the elapsed time between the start and end of a method, and the CPU time is the total time a method uses on each core of the CPU. For a single thread method, the running time is exactly the same as the CPU time. And for a multi-threading method running on a multi-core CPU, the running time is typically shorter than the CPU time. RNA-Skim is submitted as a single thread method. Sailfish, Cufflinks with TopHat as the aligner, and RSEM with Bowtie as the aligner are submitted with multi-threading enabled and requiring 8 threads. eXpress is an online algorithm, and it can quantify an streaming input of alignments generated by Bowtie in real-time. Bowtie and eXpress use 6 and 2 threads for alignment and quantification respectively.

Table 3.3 summarizes the running time of all five methods. RNA-Skim is the fastest, about 11 times faster than the second best method, Sailfish, on the CPU time. Even when Sailfish uses 8 threads, RNA-Skim is about 1.6 times faster on the running time by just using one thread. Since the aligner usually consumes lots of computation time, RNA-Skim has more than 100 times speedup on the CPU time compared with Cufflinks, RSEM and eXpress.

Overall, these results demonstrate that RNA-Skim provides comparable accuracy with other methods on both simulated and real data, using a much shorter running time.

3.6 Discussion and Conclusion

I introduced RNA-Skim, a lightweight method that can rapidly and efficiently estimate the transcript abundance levels in RNA-Seq data. RNA-Skim exploits the property of sig-mers, significantly reducing the number of k-mers used by the method and the scale of the optimization problem solved by the EM algorithm. Based on the benchmark, it is at least 10 times faster than any alternative methods. To the best of my knowledge, the design principle of almost all existing methods is to use as much data as possible for RNA-Seq quantification. The results are encouraging, in the sense that it demonstrates a different yet promising direction of building a much faster method by discovering and using only informative and reliable features — the counts of sig-mers in RNA-Seq data.

Currently, the annotation databases are incomplete and still in bootstrapping. Aligners and alignment-dependent RNA-Seq methods are commonly used to allow unknown transcript discovery, which will further improve the completeness and accuracy of the annotation databases. The performance of tools like Sailfish and RNA-Skim depends on the quality of the annotation database. Their accuracy is likely to improve in both accuracy and efficiency when better annotation databases become available.

CHAPTER 4

HTREEQA: USING SEMI-PERFECT PHYLOGENY TREES IN QUANTITATIVE TRAIT
LOCI STUDY ON GENOTYPE DATA

In previous two chapters, I've discussed possible approaches to improve the accuracy and the efficiency of the computational tools for RNA-Seq analysis. In this chapter, I focus on a phylogeny-based method for QTL mapping for finding the associations between the genetic variants and the phenotypes. These phenotypes can be either RNA-Seq based transcript abundances (e.g., generated by RNASkim) or common quantitative traits such as height and susceptibility to common diseases.

4.1 Introduction

A common analytic application in multiparent populations is the detection of statistically significant association between genetic variants and phenotypes (Aylor et al., 2011b; Kelada et al., 2012; Ferris et al., 2013; Phillippi et al., 2014). Many existing QTL mapping methods consider each genetic marker independently (Akey et al., 2001; Pe'er et al., 2006; Thomas, 2004). Standard statistical tests (such as the F-test) are used to measure the significance of association between a phenotype and every SNP in the genome. These single marker-based methods usually do not consider the effects of (both genotyped and ungenotyped) neighboring markers and hence may fail to discover QTLs for complex traits. To address this limitation, cluster-based methods, such as HAM (McClurg et al., 2006), QHPM (Onkamo et al., 2002) and HapMiner (Li and Jiang, 2005), have been developed. Typically the genome is partitioned into a series of intervals. For each interval these methods first cluster samples based on the genotypes within it, and then assess the statistical correlation between the clusters and the phenotype of interest. The result is sensitive to the granularity of the partition, the definition of genotype similarity, and the choice of clustering algorithms. More importantly, these methods tend to assume mutations are the only events that cause the differences in the DNA sequences of the samples, although this may not fully represent the genetic background underlying the differences.

Phylogeny trees have been widely used to model evolutionary history among different species, subspecies or strains (Yang et al., 2011). Their use in association study requires inferring an accurate global phylogeny tree from the DNA sequences (Larribe et al., 2002; Morris et al., 2002; Minichiello and Durbin, 2006). This may not be feasible for the high density markers in current QTL analysis. Some recent methods, such as Genomic Control (Devlin and Roeder, 1999), EIGENSTRAT (Price et al., 2006), and EMMA (Kang et al., 2008), attempt to build global models to account for genetic effects. EMMA computes a kinship matrix in order to correct the effect of the population structure. Genomic Control estimates an inflation factor of the test statistics to account for the inflation problem caused by unbalanced population structure. EIGENSTRAT performs an orthogonal transformation on the genotypes using principal component analysis (PCA) and then conducts the association study in the transformed space. However, the genetic background of the samples may not always be adequately captured by a global model. This is particularly true for some multiparent crosses. For example, the incipient Collaborative Cross population (Pre-CC). There is no significant global population stratification among the Pre-CC lines since each of the eight founders contributes roughly one-eighth of their entire genome (Aylor et al., 2011a). This unique design removes the need for global population structure correction in QTL mapping.

However, some *local* population structure may still exist. Because of the limited number of recombinations occurred since the founder generation, the genome of each CC line is a coarse mosaic of composed segments from the eight founders. In a genomic region, a CC line may be determined by the contribution from a single founder and none from the rest. Since the eight founders are from three subspecies, local population structure may exist in these CC lines. Uneven genetic background are observed at the chromosome level in the 184 genotyped Pre-CC lines, and such pattern only becomes stronger when at finer resolutions. (Please see the section on Results and Discussion for further discussion of the local population structure in the Pre-CC lines).

Local phylogeny becomes a natural choice for capturing this type of effect. Several recent methods (e.g., TreeLD (Zöllner and Pritchard, 2005), TreeDT (Sevon et al., 2006), Blossoc (Mailund et al., 2006; Besenbacher et al., 2009), and TreeQA (Pan et al., 2008, 2009)) have adopted local perfect phylogeny trees to model the genetic distance between samples. These methods examine possible groupings induced by each local phylogeny and report the ones showing strong statistical associations with the phenotype. Since these methods require a large number of statistical tests and often large permutation tests, they are prone to multiple testing errors and incur significant computational burden. TreeLD and TreeDT can handle only a very small number of SNP markers and thus they are not suitable for large scale QTL mapping. Blossoc is more efficient and can process the entire genome but still needs days to perform a large number of permutation tests. The recently proposed TreeQA algorithm utilizes several effective pruning techniques to reduce computational burden and is able to finish large permutation tests in a few hours.

A common limitation shared by all of these local phylogeny-based methods is that the perfect phylogeny trees can be only constructed from haplotypes. These methods either assume that samples are inbred (i.e., no heterozygosity) which is not true for many large mammalian multiparent crosses including the Pre-CC lines, or that a preprocessing step *phases* each genotype into a pair of haplotypes. However, haplotype reconstruction itself is a non-trivial process that is both time consuming (Scheet and Stephens, 2006) and error prone (Ding et al., 2008). Even if haplotypes are phased accurately, the two haplotypes of the same sample may be located at different branches of a phylogeny tree and will be treated as if they were independent samples in subsequent statistical tests. This may create a bias favoring additive effects and lead to spurious results. For example, consider a recessive phenotype, A/a are used to represent the majority and minority alleles at the causative locus. The local phylogeny tree built from the surrounding region has an edge corresponding to the causative SNP that separates the samples into two groups carrying A and a alleles respectively. Each heterozygous A/a sample is phased into two haplotypes, each of which belonging to a different group. The group having allele a would have mixed phenotypes. This may weaken the power of any statistical tests and fail to detect the causative edge (Wang and Sheffield, 2005; Lettre et al., 2007). The scenario may become even worse for phenotypes having overdominant effects on heterozygous samples.

Therefore, a natural question to ask is whether a phylogeny-based QTL mapping can be used on unphased genotypes directly. In this chapter, I introduce the model of *tri-state semi-perfect phylogeny tree* directly built from unphased genotype data, and explore its utility in GWAS. This chapter introduces HTreeQA (Zhang et al., 2012), which has all the advantages of phylogeny-based methods and does not require a separate phasing step. I also demonstrate via simulation studies that HTreeQA can detect a wider range of genetic effects than other alternative methods.

4.2 Method

4.2.1 Notations

I follow the convention of using primed notation for unphased genotype data. Suppose that there are m individuals and n SNPs. $\{S'_1, S'_2, \cdots, S'_n\}$ are used to represent the unphased SNPs and $\{S_1, S_2, \cdots, S_n\}$ to represent the phased SNPs. The unphased genotypes can be represented as an $m \times n$ matrix $\mathcal{M}'$, where the k-th row corresponds to the genotype of the k-th individual and the l-th column corresponds to the l-th SNP marker S'_l. Similarly, the $2m$ haplotypes can be represented as a $2m \times n$ matrix $\mathcal{M}$, where the $2k$-th and $(2k+1)$-th rows correspond to the haplotypes of the k-th individual. In the haplotype matrix $\mathcal{M}$, '0' and '1' are used to represent the major allele and the minor allele of a SNP respectively. In the genotype matrix $\mathcal{M}'$, '0', '1', and 'H' are used to represent the homozygous major allele, the homozygous minor allele, and the heterozygous allele of a SNP respectively. Table 4.1(a) shows an unphased genotype matrix, and Table 4.1(b) shows a phased haplotype matrix.

4.2.2 Perfect Phylogeny Tree

An *interval* along the genome consists of a set of consecutive SNPs. It corresponds to a submatrix $C_{u,v}(\mathcal{M})$ of $\mathcal{M}$ that contains all columns between the u-th column and the v-th column. A *perfect phylogeny tree* is the tree representation of the evolution genealogy for an interval in the genome (Gusfield, 1991).

Definition 3 *Given an interval $C_{u,v}(\mathcal{M})$ of $2m$ haplotypes and n SNPs, a perfect phylogeny tree is a tree in which the haplotype sequences are the leaves and SNPs are the edges. Given an allele of any SNP, the subgraph induced by all the nodes that carry the same allele is still a connected subtree.*

(a) The unphased genotype matrix

Sample ID	X_1	X_2	X_3	X_4	X_5	Phenotype
A	0	0	1	1	0	10
B	0	0	1	0	1	10
C	H	1	0	0	0	2
D	H	H	0	0	0	10
E	1	1	0	0	0	2

(b) The phased haplotype matrix

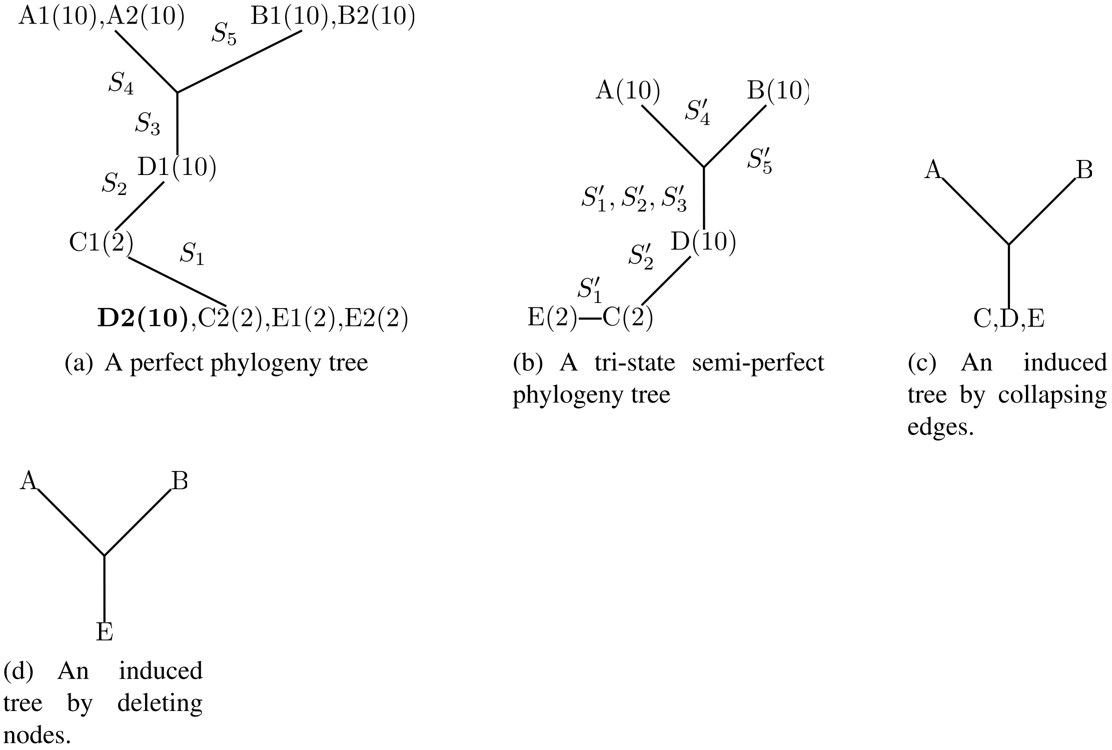
Haplotype ID	S_1	S_2	S_3	S_4	S_5	Phenotype
A1	0	0	1	1	0	10
A2	0	0	1	1	0	10
B1	0	0	1	0	1	10
B2	0	0	1	0	1	10
C1	0	1	0	0	0	2
C2	1	1	0	0	0	2
D1	0	0	0	0	0	10
D2	1	1	0	0	0	10
E1	1	1	0	0	0	2
E2	1	1	0	0	0	2

(c) The transformed genotype matrix. Bold columns are selected for building the tri-state semi-perfect phylogeny tree

ID	$S'_1(0)$	$S'_1(1)$	$S'_1(H)$	$S'_2(0)$	$S'_2(1)$	$S'_2(H)$	$S'_3(0)$	$S'_3(1)$	$S'_3(H)$	$S'_4(0)$	$S'_4(1)$	$S'_4(H)$	$S'_5(0)$	$S'_5(1)$	$S'_5(H)$
A	**0**	**0**	0	**0**	**0**	0	**1**	1	**0**	**1**	1	**0**	**0**	0	**0**
B	**0**	**0**	0	**0**	**0**	0	**1**	1	**0**	**0**	0	**0**	**1**	1	**0**
C	**1**	**0**	1	**1**	**1**	0	**0**	0	**0**	**0**	0	**0**	**0**	0	**0**
D	**1**	**0**	1	**1**	**0**	1	**0**	0	**0**	**0**	0	**0**	**0**	0	**0**
E	**1**	**1**	0	**1**	**1**	0	**0**	0	**0**	**0**	0	**0**	**0**	0	**0**

Table 4.1: An example data set

The perfect phylogeny can be treated as an evolutionary history for the interval. Each edge represents the mutation event that derives two alleles of the corresponding SNP. All the haplotypes can be explained by the the evolutionary history without any recombination event. For example, Figure 4.1(a) shows the perfect phylogeny tree built from the haplotypes in Table 4.1(b).



(a) A perfect phylogeny tree

(b) A tri-state semi-perfect phylogeny tree

(c) An induced tree by collapsing edges.

(d) An induced tree by deleting nodes.

Figure 4.1: Figure 4.1(a) is the perfect phylogeny tree generated on the phased haplotypes in Table 4.1(b). Each node is labeled by its haplotype ID followed by the corresponding phenotype value. Figure 4.1(b) is a tri-state semi-perfect phylogeny tree generated on the unphased genotypes in Table 4.1(a). Each node is labeled by its sample ID followed by the corresponding phenotype value. Figure 4.1(c) is the corresponding perfect phylogeny tree by deleting S'_1 and S'_2 in Table 4.1(a). Figure 4.1(d) is the corresponding perfect phylogeny tree by deleting samples C and D in Table 4.1(a).

4.2.3 Compatible Interval

An interval $C_{u,v}(\mathcal{M})$ is a *compatible interval* if every pair of SNP markers in the interval pass the 4-gamete test (hudson and kaplan, 1985). That is, at most 3 out of the 4 possible allele pairs $\{00, 01, 10, 11\}$ appear in each pair of SNPs in the interval. This implies the existence of an evolution genealogy that can explain the evolutionary history of these two markers without recombination events, given the assumption of an infinite site model (i.e., no homoplasy). For a given interval, a perfect phylogeny exists if and only if the interval is a compatible interval. If a compatible interval is not a sub-interval of another compatible interval, it is called a *maximal* compatible interval.

4.2.4 Tri-State Semi-Perfect Phylogeny Tree

The multi-state perfect phylogeny tree (Gusfield, 2009) is a natural extension of the perfect phylogeny tree discussed above. It was originally proposed to model the rare events having multiple mutations at a single locus. Because the perfect phylogeny cannot handle heterozygous site properly, a novel utility of the multi-state phylogeny is proposed in modeling heterozygosity in QTL mapping. By treating the heterozygous allele as the third status, a tri-state phylogeny tree can be generated from a set of unphased genotypes. Since this third state is not a result of a single mutation, the tri-state phylogeny tree is a relaxation of a perfect phylogeny tree.

Definition 4 *Given an interval $C_{u,v}(\mathcal{M}')$ of m genotypes and n SNPs, a tri-state semi-perfect phylogeny tree is a tree in which the genotype sequences are the leaves and SNPs are the edges. A SNP corresponds to an edge if only two of the three possible alleles are observed, and corresponds to two edges if all three alleles are observed. Given an allele of any SNP, the subgraph induced by all the nodes that carry the same allele is still a connected subtree.*

4.2.5 Compatibility Test on Genotype Data

Given an interval $C_{u,v}(\mathcal{M})$ in the genotype matrix, a binary matrix $\overline{C_{u,v}(\mathcal{M}')}$ is constructed. Each column S_i' in $C_{u,v}(\mathcal{M})$ corresponds to three binary columns $S_i'(0)$, $S_i'(1)$ and $S_i'(H)$ in $\overline{C_{u,v}(\mathcal{M}')}$. $S_i'(0)$ is generated from S_i' by replacing every 'H' in S_i' by '1'. $S_i'(1)$ is generated from S_i' by replacing every 'H' in S_i' by '0'. $S_i'(H)$ is generated from S_i' by replacing every 'H' in S_i' by '1', and '0' and '1' in S_i' by '0'. This is equivalent to representing the '0','1','H' alleles in the heterozygous S_i' by triplets (0,0,0), (1,1,0) and (1,0,1), respectively. For example, Table 4.1(c) shows the generated binary matrix $\overline{C_{u,v}(\mathcal{M})}$ for the genotype matrix $C_{u,v}(\mathcal{M})$ in Table 4.1(a). Note that all states in $\overline{C_{u,v}(\mathcal{M})}$ are identical to that in $C_{u,v}(\mathcal{M}')$ except the 'H' alleles and $S'(H)$ columns. Given an interval, the following theorem states the necessary and sufficient condition for the existence of a tri-state semi-perfect phylogeny (dress and steel, 1992).

Theorem 5 *Given an interval $C_{u,v}(\mathcal{M}')$ in the genotype matrix, there exists a tri-state semi-perfect phylogeny, if and only if there exists a submatrix $\mathcal{S}$ formed by selecting two of the three columns in $\overline{C_{u,v}(\mathcal{M}')}$ for each SNP marker, and any pair of columns in $\mathcal{S}$ pass the 4-gamete test.*

An integer linear programming approach (Gusfield, 2009) can be used to determine whether an interval is compatible and to compute the submatrix $\mathcal{S}$. For example, in the matrix $\overline{C_{u,v}(\mathcal{M}')}$ shown in Table 4.1(c), the columns selected for $\mathcal{S}$ are highlighted in bold. Once $\mathcal{S}$ is computed, a tri-state semi-perfect phylogeny tree can be constructed by applying any standard perfect phylogeny tree algorithm on $\mathcal{S}$. For example, Figure 4.1(b) shows the tri-state semi-perfect phylogeny tree constructed from the matrix $\mathcal{S}$ in Table 4.1(c).

If there is not an heterozygous allele, each genotype will be composed of two identical haplotypes; the tri-state semi-perfect phylogeny tree is identical to the perfect phylogeny tree constructed on the haplotypes. If there are some heterozygous genotypes, removing the rows or columns in the matrix containing the heterozygous alleles does not affect the remaining part of the phylogeny tree. The tree in Figure 4.1(c) shows the perfect phylogeny tree constructed on S'_3, S'_4, S'_5 in Table 4.1(a), which can also be derived by collapsing the three edges labeled by S'_1 or S'_2 in Figure 4.1(b). If nodes C and D (that have heterozygous genotypes) are removed in Figure 4.1(b), the resulting tree is also identical to the perfect phylogeny tree constructed on A, B, E (Figure 4.1(d)). Any heterozygosity only introduces local variations in a phylogeny tree.

Another important observation can be made by comparing the perfect phylogeny tree constructed on the haplotypes to the genotype matrix. When the genotype matrix contains a small percentage of heterozygosity, the tri-state semi-perfect phylogeny tree shares a substantial common structure with the perfect phylogeny tree on the haplotypes. Figure 4.1(a) shows the perfect phylogeny tree constructed on the haplotypes in Table 4.1(b). Note that the two haplotypes (e.g., D1, D2) of the same genotype (e.g., D) may be associated with different nodes in the tree. This decoupling weakens the power of detecting non-additive genetic effects (more details later). However, this tree shares common induced subtrees with the tri-state semi-perfect phylogeny tree. Removing the nodes associated with the decoupled haplotypes will result in Figure 4.1(d), while collapsing edges connecting these nodes will result in Figure 4.1(c).

4.2.6 Phylogeny Tree based Test

An edge in a phylogeny tree connects two disjoint subtrees. Removing x edges partitions the tree into $x + 1$ subtrees. For example, removing the two edges labeled with S'_1 and S'_2 in Figure 4.1(b) partitions genotypes into three groups { A, B, D }, { C }, and { E }.

The statistical correlation between a partition and the phenotype can be examined by the F-statistics. Assuming that for a total of t individuals, there are p groups, and the ith group contains t_i individuals. X_{ij} represents the ith element in the jth group, $\overline{X}_j$ to represent the mean of the jth group, and $\overline{X}$ to represent the overall mean value. Given such a grouping of phenotype values, G, the F-statistics is defined as

$$F(G) = \frac{\sum_{j=1}^{p} t_j(\overline{X}_j - \overline{X})^2}{\sum_{j=1}^{p} \sum_{i=1}^{t_j} (X_{ij} - \overline{X}_j)^2} \tag{4.1}$$

The corresponding p-value of $F(G)$ can be calculated in the following way. If the phenotype values from each group follow a normal distribution, an F-test is applied to obtain the corresponding p-value. Otherwise, a permutation test is needed. The p-value is defined as $\frac{n}{nPerm}$ where $nPerm$ is the number of permutations and n is the number of times when the F-statistics of the permuted phenotype is larger than $F(G)$.

HTreeQA examines all possible partitions generated by removing edges in the tree. The partition that generates the most significant p-value is reported. The corresponding p-value is used as the nominal (uncorrected) p-value of the association between the compatible interval and the phenotype.

4.2.7 Permutation Test for Family-Wise Error Rate Controlling

Appropriate multiple testing correction is crucial for QTL studies. In HTreeQA, the widely used permutation test is applied to control family-wise error rate (Westfall and Young, 1993; Churchill and Doerge, 1994). In each permutation, the phenotype values are randomly shuffled and reassigned to individuals. For each permuted phenotype, HTreeQA repeats the previously described procedure and find the smallest p-value. The corrected p-value is the proportion of the permuted data whose p-values are more significant than that of the original data. The corrected p-value is referred as the permutation p-value.

The basic routine of HTreeQA is summarized in Figure 4.2.

4.2.8 Comparison between TreeQA and HTreeQA

Two alternative approaches for local phylogeny-based QTL mapping methods are outlined here, and I also discuss their pros and cons.

- HTreeQA: compatible intervals are computed using integer linear programming

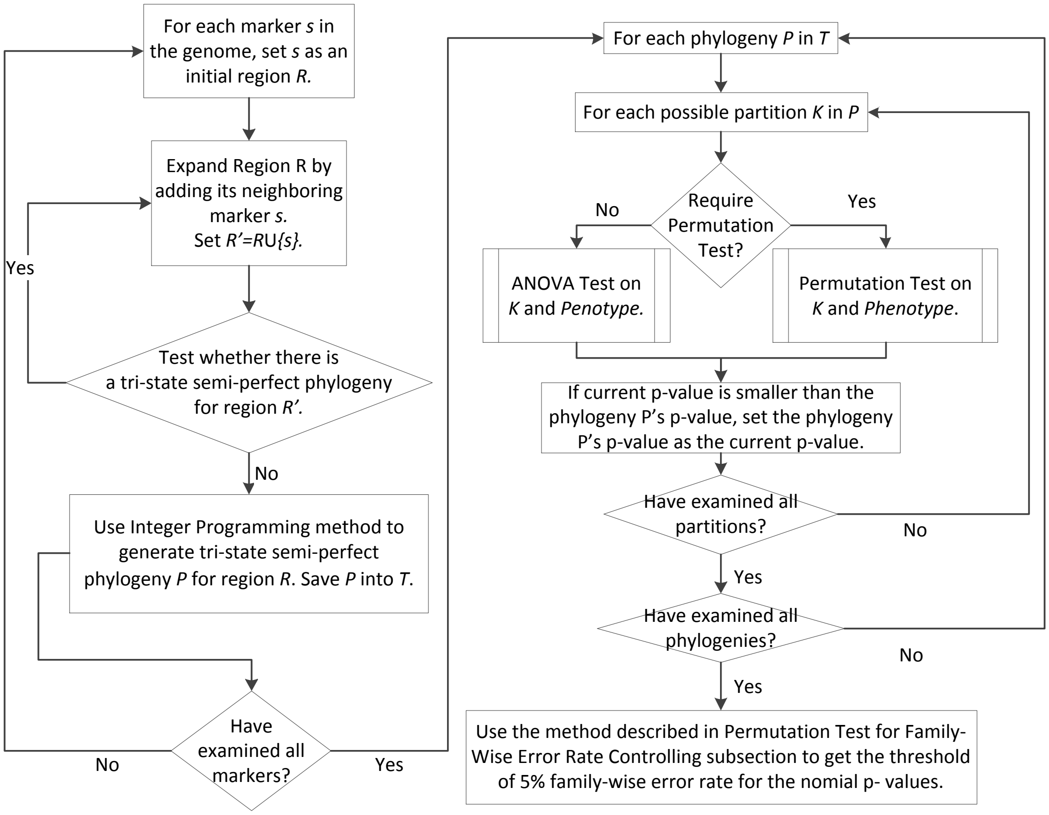

For each marker *s* in the genome, set *s* as an initial region *R.*
Expand Region R by adding its neighboring marker *s.* Set *R'=RU{s}.*
Yes
Test whether there is a tri-state semi-perfect phylogeny for region *R'.*
No
Use Integer Programming method to generate tri-state semi-perfect phylogeny *P* for region *R*. Save *P* into *T*.
Have examined all markers?
No
Yes
For each phylogeny *P* in *T*
For each possible partition *K* in *P*
Require Permutation Test?
No
Yes
ANOVA Test on *K* and *Penotype.*
Permutation Test on *K* and *Phenotype*.
If current p-value is smaller than the phylogeny P's p-value, set the phylogeny P's p-value as the current p-value.
Have examined all partitions?
No
Yes
Have examined all phylogenies?
No
Yes
Use the method described in Permutation Test for Family-Wise Error Rate Controlling subsection to get the threshold of 5% family-wise error rate for the nomial p- values.

Figure 4.2: The workflow of HTreeQA. The input are the genotype and phenotype data. The output is a list of phylogenies and their p-values for measureing the association with the phenotype, and a threshold of p-value representing the 5% family-wise error rate (FWER).

and construct a tri-state semi-perfect phylogeny trees for each compatible interval. Then HTreeQA is applied to find significant associations.

- Running TreeQA on phased data: the genotypes are first phased using any standard phasing algorithm and then TreeQA is applied on the resulting haplotypes. Each haplotype is assumed to have the same phenotype value as the original genotype.

The second approach has an inherent drawback. It decouples the two haplotypes of the same genotype. As a result, the two haplotypes may reside in remote branches of the tree, which limits the ability to test certain genetic effects in QTL mapping. For example, the phenotype in Table 4.1(a) follows a recessive model defined on S'_2: The phenotype is 2 for samples (C,E) having minor allele ('1') and is 10 for the remaining samples A, B, D (with alleles '0' or 'H'). There does not exist a set of edges in Figure 4.1(a) that can perfectly separate these two groups. (The haplotype D2 will always be in the same group as C1, E1, E2.) In contrast, the tri-state semi-perfect phylogeny tree has an edge S'_2 that perfectly separates A, B, D from C, E. Therefore, the tri-state semi-perfect phylogeny tree is more suitable for handling heterozygosity in association studies.

4.3 Materials

4.3.1 Collaborative Cross

The Collaborative Cross ("Collaborative Cross Consortium", 2012) is a large panel of recombinant inbred multiparent crosses bred from a set of 8 inbred founder mouse strains (short names in parentheses): 129S1/SvlmJ (129S1), A/J (AJ), C57BL/6J (B6), NOD/ShiLtJ (NOD), NZO/HILtJ (NZO), CAST/EiJ (CAST), PWK/PhJ (PWK), and WSB/EiJ (WSB). Breeding of the CC is an ongoing effort, and at present a relatively small number of finalized lines are available. Nonetheless, partially inbred lines taken from an early stage of the CC breeding process, the so-called incipient strains of Collaborative Cross (Pre-CC) population, has been studied and used for QTL identification (Aylor et al., 2011b; Kelada et al., 2012; Ferris et al., 2013; Phillippi et al., 2014). This comprises 184 lines, each with one replicate, that have attained on average 6.7 generation of inbreeding following the initial 8-way cross, resulting in genomes with approximately 16% residual heterozygosity. The genotypes at approximately 180K SNPs are collected using the mouse diversity array (Yang et al., 2009), which can be accessed through the CC status website (http://csbio.unc.edu/CCstatus/index.py). Two phenotypes are studied. One is the white head spot, which was originally observed on one of the CC founders, WSB/EiJ. Because there are no white head-spotted mice found in F1 crosses of the CC founders, the phenotype is believed to be a recessive trait. Among the 184 mice, there are six with white head spot. Another phenotype I study is the average daily running distance for mice of 5-6 days old. This is a typical measurement for mouse activity. The phentotypes are supplied as supplementary materials.

4.3.2 Synthetic Data Sets

The phenotype was simulated using 3 different models of genetic effects: additive, recessive, and overdominant (a special case of epistasis effect) models. The overdominant model is also included because I observe that heterozygous individuals sometimes exhibit extreme phenotypes. This phenomenon cannot be captured by an additive or recessive model.

To simulate phenotypes, I adopt the method used in (Long and Langley, 1999). To simulate an additive phenotype for a given SNP, the following formula is used:

$$y_i = \sqrt{1-\pi}\mathrm{N}(0,1) + Q_i\sqrt{\frac{\pi}{2p(1-p)}}$$

where π is the percentage of the variation attributable to the quantitative trait nucleotide (QTN), $\mathrm{N}(0,1)$ is the standard normal distribution, and p is the minor allele frequency. In the additive model, Q_i takes values -1, 0 and 1 for homozygous wild type, heterozygous type, or homozygous type, respectively. For recessive and overdominant models, the following formula is used,

$$y_i = \sqrt{1-\pi}\mathrm{N}(0,1) + Q_i'\sqrt{\frac{\pi}{2p'(1-p')}}$$

where p' is the fraction of individuals that are homozygous mutants. In a recessive model, Q_i' is 1 for homozygous mutant and 0 otherwise. In an overdominant model,

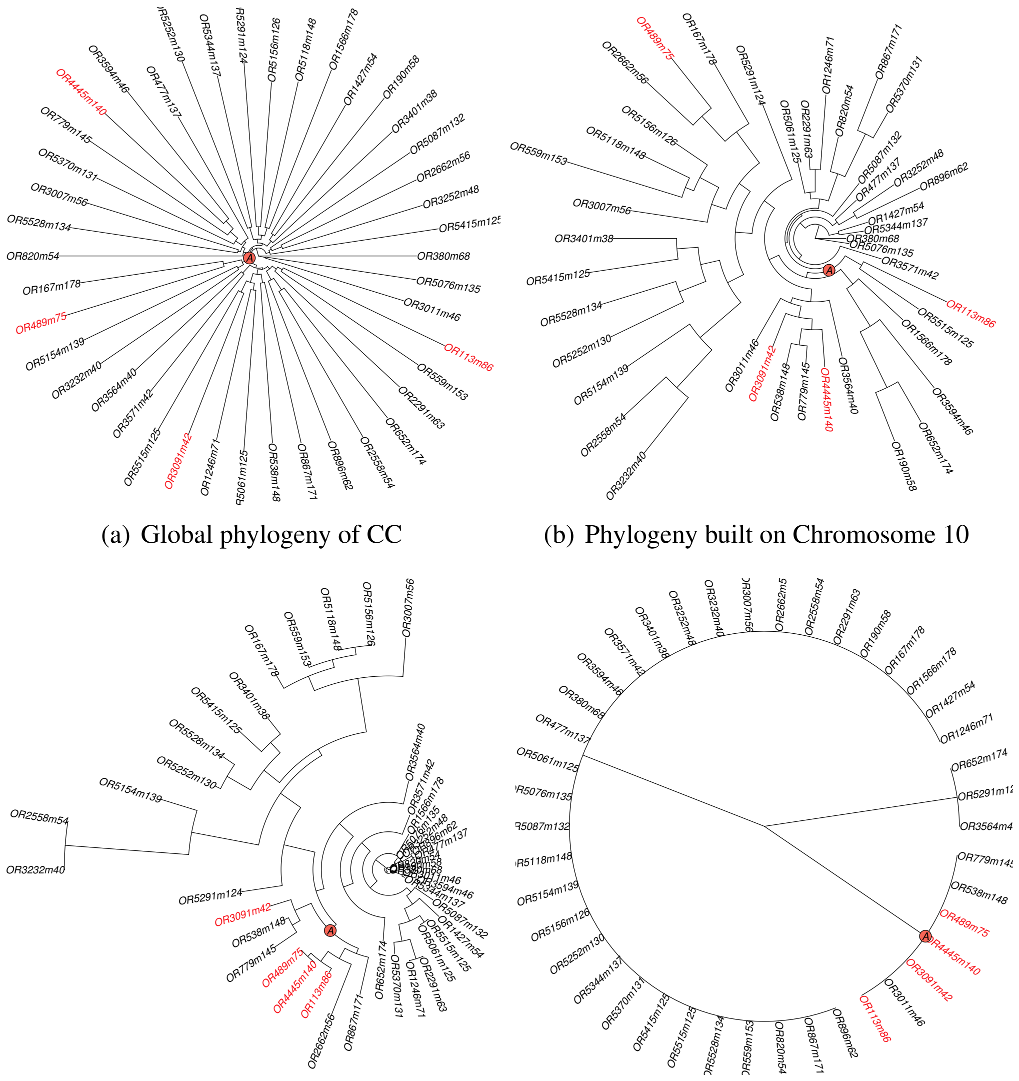
Q_i takes 1 for heterozygous mutant and 0 otherwise. All causative SNPs are removed from the genotypes prior to analysis. I represent results of a wide range of realistic contributions of genetic variations by testing five genetic variation settings of π: 0.05, 0.1, 0.15, 0.2 and 0.25.

Genotypes of 170 independent individuals are simulated. Under each genetic effect model, 100 independent test cases under each setting are generated. In each case, there are 10000 SNPs and one causative SNP is randomly picked among the SNPs with Minor Allele Frequency (MAF) larger than 0.15.

4.4 Results and Discussion

4.4.1 Population Structure in the Pre-CC Lines

Population stratification is an important issue in QTL analysis. Spurious associations may be induced by the stratification if it is not addressed properly (Kang et al., 2008). The combinatorial breeding design of the Collaborative Cross yields genetically independent incipient CC lines, that ensures balanced contributions of all eight founder strains without noticeable global population stratification (Aylor et al., 2011a). Figure 4.3(a) shows a global phylogeny tree of 43 randomly selected Pre-CC lines. The balanced tree structure illustrates that these mice are genetically diverse and equally distant from each other. This observation is further confirmed by the kinship matrix in Figure 4.4(a) used by EMMA for modeling genetic background (Kang et al., 2008). In Figure 4.4(a), each row (column) of the kinship matrix corresponds to a CC strain. Each entry in the matrix is the kinship coefficient that represents the genetic relatedness between the two mice. I observe that all off-diagonal entries in Figure 4.4(a) have almost identical values (around 0.8), which suggests that no significant global population stratification exists in these Pre-CC mice.



(a) Global phylogeny of CC

(b) Phylogeny built on Chromosome 10

(c) Phylogeny built on an interval from 85Mbps to 105Mbps on Chromosome 10

(d) Tri-state semi-perfect phylogeny built on the compatible interval (20Kbps) reported as a QTL of white spot

Figure 4.3: Four phylogenies of 43 randomly selected (from a total of 184) Pre-CC mice. The sum of the edge depth between a leaf and the origin represents the genetic distance of the corresponding mouse from the common ancestry of the 43 mice. The mice with white head spot are highlighted in red. Their nearest common ancestor is indicated by a circled "A" in each figure. In Figure 4.3(a), the global phylogeny is balanced and all mice are almost equally distant from each other. The phylogenies in Figure 4.3(b) and 4.3(c) are no longer balanced, with several deep branches. The local population structure is a confounding factor that complexes the QTL analysis. The tri-state semi-perfect phylogeny in Figure 4.3(d) has the simplest structure with an informative branch that contains all four white spot mice.

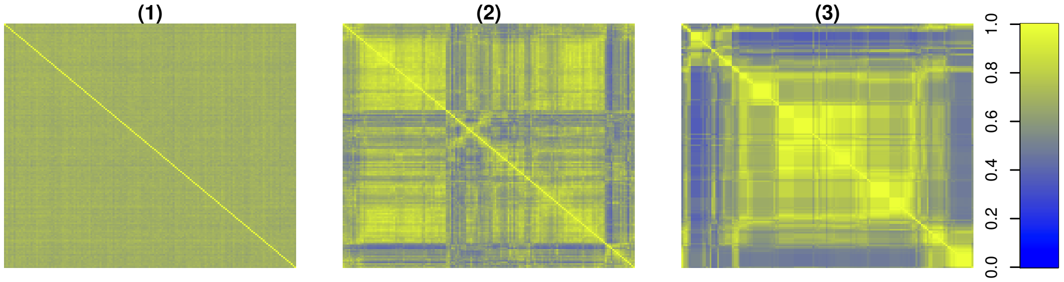


Figure 4.4: Three kinship matrices represent the genetic relatedness over the entire genome between any pair of the 184 CC mice based on the whole genome (a), the Chromosome 10 (b), and the 20Mbps interval in Chromosome 10 (c) respectively. The mice are arranged in the same order in both x and y axes. In Figure 4.4(a), all off-diagonal entries have almost identical values, suggesting that there is no global population structure. In Figure 4.4(b)(c), the mice are arranged in the order of their genetic relatedness, genetically similar mice are near each other.

4.4.2 EMMA will degenerate to standard linear model in Collaborotive Corss

EMMA can efficiently control population structure in QTL mapping, however, it becomes unnecessary when using EMMA to analyze CC genome. In this section, I provide a statistical analysis that EMMA degenerates to a standard linear model when applied to the CC lines.

First, we define a new class of matrix named $K_{\text{uniform}}(D, S)$,

$$K_{\text{uniform}}(D, S) = \begin{pmatrix} D & S & \cdots & S \\ S & D & \cdots & S \\ \vdots & \vdots & \ddots & \vdots \\ S & S & \cdots & D \end{pmatrix} \tag{4.2}$$

where D represents the diagonal entries and S represents the off-diagonal entries in the matrix.

Assume that **y** is a vector of phenotypes, **X** is a vector of fixed effects from a SNP, and **e** is a vector of residual effects for each individual. We omit the indicator matrix Z used in original EMMA model, because in the CC data, Z is an identity matrix. The EMMA model is presented in the following form:

$$\mathbf{y} = \mu\mathbf{1} + \mathbf{X}\beta + \mathbf{u} + \mathbf{e} \tag{4.3}$$

$$\mathbf{u} \sim \text{MVN}(\mathbf{0}, \sigma_K^2 K_{\text{emma}}) \tag{4.4}$$

$$\mu \sim \text{N}(0, \sigma_\mu^2) \tag{4.5}$$

$$\mathbf{e} \sim \text{MVN}(\mathbf{0}, \sigma_e^2 K_{\text{uniform}}(1, 0)) \tag{4.6}$$

where MVN represents a multivariate normal distribution. K_{emma} is the kinship matrix inferred by the EMMA package.

Similarly, a standard linear model is in the following form:

$$\mathbf{y} = \mu\mathbf{1} + \mathbf{X}\beta + \mathbf{e} \tag{4.7}$$

$$\mu \sim \text{N}(0, \sigma_\mu^2) \tag{4.8}$$

$$\mathbf{e} \sim \text{MVN}(\mathbf{0}, \sigma_e^2 K_{\text{uniform}}(1, 0)) \tag{4.9}$$

Assuming the samples of a population have exactly the same relatedness S:

$$K_{\text{uniform}}(1, S) = K_{\text{uniform}}(S, S) + K_{\text{uniform}}(1 - S, 0) \tag{4.10}$$

$$\mu\mathbf{1} \sim \text{MVN}(\mathbf{0}, \sigma_\mu K_{\text{uniform}}(1, 1)) \tag{4.11}$$

$$\mathbf{e} \sim \text{MVN}(\mathbf{0}, \sigma_\mu K_{\text{uniform}}(1, 0)) \tag{4.12}$$

Thus, if $K_{\text{emma}} = K_{\text{uniform}}(1, S)$, by re-factorization of the random effects in the EMMA model, we have

$$\mathbf{y} = \mu\mathbf{1} + \mathbf{X}\beta + \mathbf{e} \tag{4.13}$$

$$\mu\mathbf{1} \sim \text{MVN}(\mathbf{0}, K_{\text{uniform}}(\sigma_\mu^2 + \sigma_K^2 S, \sigma_\mu^2 + \sigma_K^2 S)) \tag{4.14}$$

$$\mathbf{e} \sim \text{MVN}(\mathbf{0}, \sigma_e^2 K_{\text{uniform}}((1 - \sigma_K^2)S + 1, 0)) \tag{4.15}$$

This has the same form of a standard linear regression model. In CC, the kinship matrix can be represented by a K_{uniform} matrix with tolerable numerical error. This suggests that there is no significant difference between EMMA and the standard linear regression model when these two methods are applied to Collaborative Cross data

4.4.3 Local Population Structure

Although the genome of each CC line receives a balanced contribution from each founder strain, the founder contribution is not uniformly distributed along the genome because of the small number of recombination events undergone by each CC line. The genome of a CC line is essentially a mosaic of a small number of founder haplotype segments. On average, Pre-CC autosomal genomes had 142.3 segments on average (st dev. = 21.8) with a median segment length of 10.46Mb (Aylor et al., 2011a). As a result, some local population structure may be observed because the eight founder strains are not equally distant from each other (i.e. three of founders are wild strains). The population structure is visible at the chromosome level. For example, there are several deep branches in the phylogeny tree of the selected Pre-CC mice built on Chromosome 10 (Figure 4.3(b)). The corresponding kinship matrix in Figure 4.4(b) shows that there are at least three subpopulations. The subpopulation structure is more evident if narrowed down to a 20Mbps interval from 85Mbps to 105Mbps on Chromosome 10. The phylogeny tree in Figure 4.3(c) becomes more skewed, and the corresponding kinship matrix in Figure 4.4(c) also exhibits more pronounced structural patterns.

4.4.4 Selected Methods for Comparison

HTreeQA is also compared with existing methods: TreeQA (Pan et al., 2008, 2009), BLOSSOC (Mailund et al., 2006; Besenbacher et al., 2009), EMMA (Kang et al.,

	Methods
Non-Phylogeny-based Methods	SMA, HAM, EMMA
Phylogeny-based Methods	BLOSSOC, TreeQA, HTreeQA

Table 4.2: Selected methods for comparison

2008), and HAM (McClurg et al., 2006) using both real and simulated data sets. Some other methods, such as HapMiner (Li and Jiang, 2005) and TreeLD (Zöllner and Pritchard, 2005), are too slow to process large data sets. For comparison purposes, the other two methods are also implemented: SMA (Single Marker Association Mapping) and HAM (Haplotype Association Mapping). In SMA, each SNP marker partitions samples into groups based on the alleles. The ANOVA test is used to evaluate the significance of the partition. In HAM, a sliding window of 3 consecutive SNP is used to group samples based on their sequences, and the ANOVA test is conducted to test the association between the phenotypes and the grouping. FastPhase (Scheet and Stephens, 2006) is used to reconstruct haplotypes from the genotypes for the methods that require haplotype data (TreeQA and BLOSSOC).

Note that BLOSSOC, TreeQA, and HTreeQA are phylogeny-based methods. SMA, HAM, and EMMA are non-phylogeny based methods. Although EMMA offers an option to use global phylogeny to estimate the kinship matrix, it does not test the associations between the phenotype and the phylogenetic trees. Table 4.2 shows the selected methods for comparison.

4.4.5 Performance Comparison on the White Head Spot Phenotype

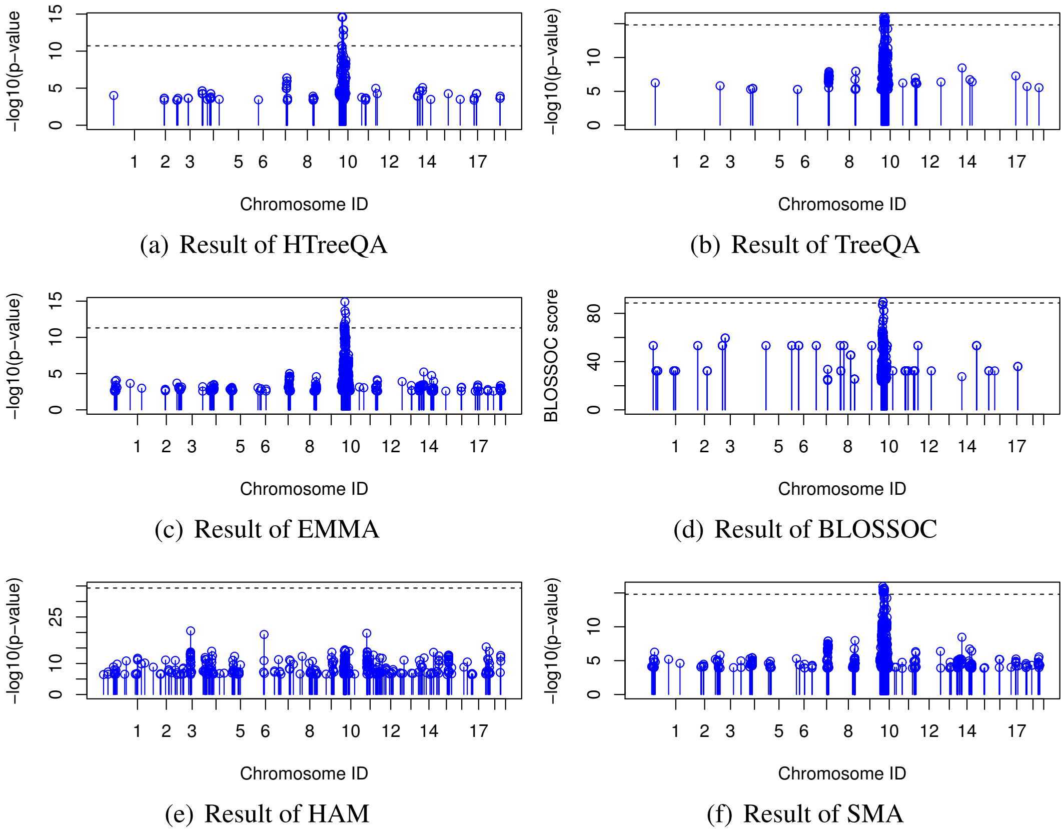
The white head spot is known as a recessive trait carried by WSB/EiJ (Aylor et al., 2011a). I apply the selected methods to the white head spot phenotype. A permutation test is applied to control the family-wise error rate (FWER) (Westfall and Young, 1993; Churchill and Doerge, 1994). With FWER = 0.05, all the selected methods except

HAM identify a QTL, which is around 100Mbps in Chromosome 10 (Figure 4.5). This QTL is close to a gene named *kit ligand* known to be controlling white spotting (Aylor et al., 2011a). HAM fails to detect the QTL because it does not consider the compatibility between consecutive SNPs. The incompatibility between two consecutive SNPs suggests a high possibility of having a historical recombination event between them. Treating an interval containing incompatible SNPs as a single locus may lead to spurious results. The phylogeny-based methods including HTreeQA can avoid this problem by only examining phylogeny trees constructed from compatible intervals.

In each figure of Figures 4.3(a)-4.3(d), the nearest common ancestor of the four white head spot mice (highlighted in red) is marked by a circled "A". I observe from Figures 4.3(a)-4.3(c) that the distance between the common ancestor and the four mice becomes smaller when the interval on which the tree is built becomes shorter. It is evident that the four white spot mice are clustered in the phylogeny tree built over the 20Mb region in Figure 4.3(c), despite the (local) population structure. This becomes more clear in Figure 4.3(d) where the four white head spot mice having white head spot located on the same branch of the tri-state semi-perfect phylogeny tree built on the compatible interval at the QTL. This demonstrates the effectiveness of the proposed model.

4.4.6 Performance Comparison on the Mouse Running Distance Phenotype

I apply the selected methods on the phenotype "Mouse Running Distance at day 5/6". With FWER=0.05, all the methods except SMA, identified a QTL at 169Mbp-169.2Mbp (89cM) on Chromosome 1 as shown in Figure 4.6. The QTL falls into the previously reported *cplaq3* region (Mayeda and Hofstetter, 1999). A later study also confirmed this QTL (Hofstetter et al., 2003).



(a) Result of HTreeQA

(b) Result of TreeQA

(c) Result of EMMA

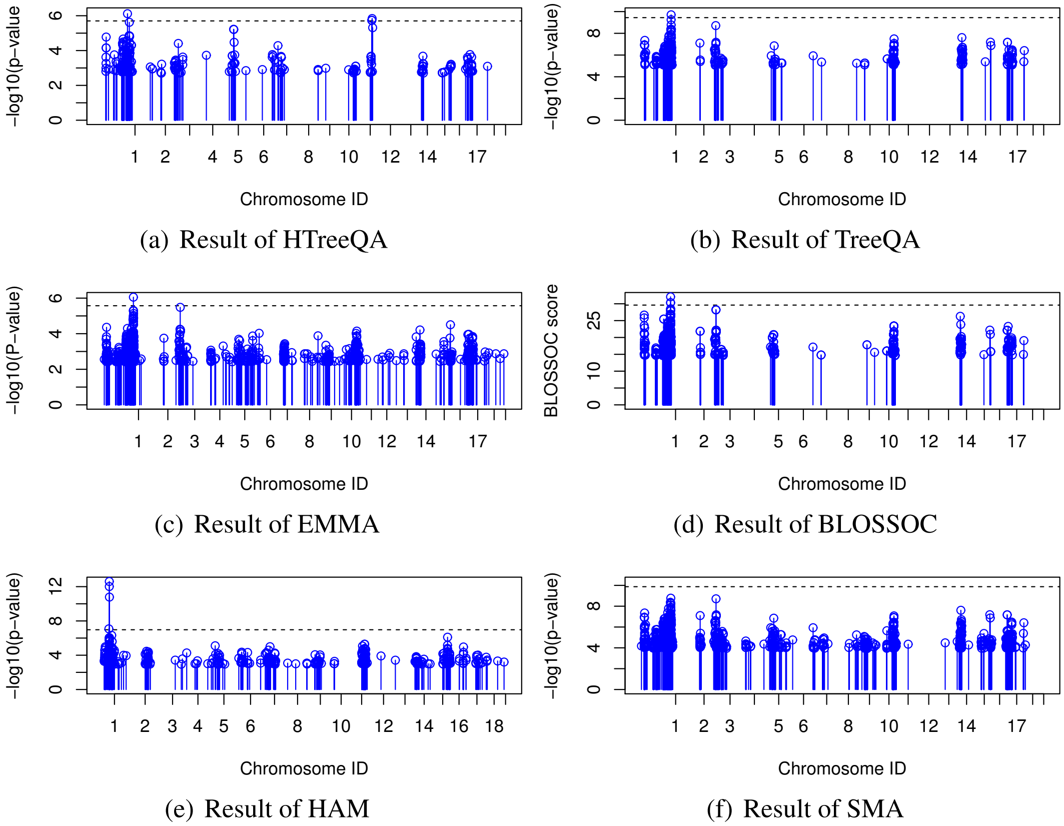
(d) Result of BLOSSOC

(e) Result of HAM

(f) Result of SMA

Figure 4.5: QTL mapping of the white head spot phenotype. Only the SNPs that have top 0.5% -log(p-value) or BLOSSOC score are plotted in the figure. One QTL is detected by HTreeQA, which is near the location of gene *kit ligand*. The remaining methods except HAM have similar results to that of HTreeQA. The dashed line is the significance level FWER = 0.05.

Among the selected methods, only HTreeQA identified another QTL with FWER=0.05, in the region of 16M-25Mbps (8-12.5cM) on Chromosome 12. The QTL falls into an unnamed QTL region at 11cM on Chromosome 12 reported in (Hofstetter et al., 2003). The reason that many methods fail to report this QTL is that these methods have limited power in detecting non-additive effects. This result demonstrates that HTreeQA can detect more types of effects than the other methods.



(a) Result of HTreeQA

(b) Result of TreeQA

(c) Result of EMMA

(d) Result of BLOSSOC

(e) Result of HAM

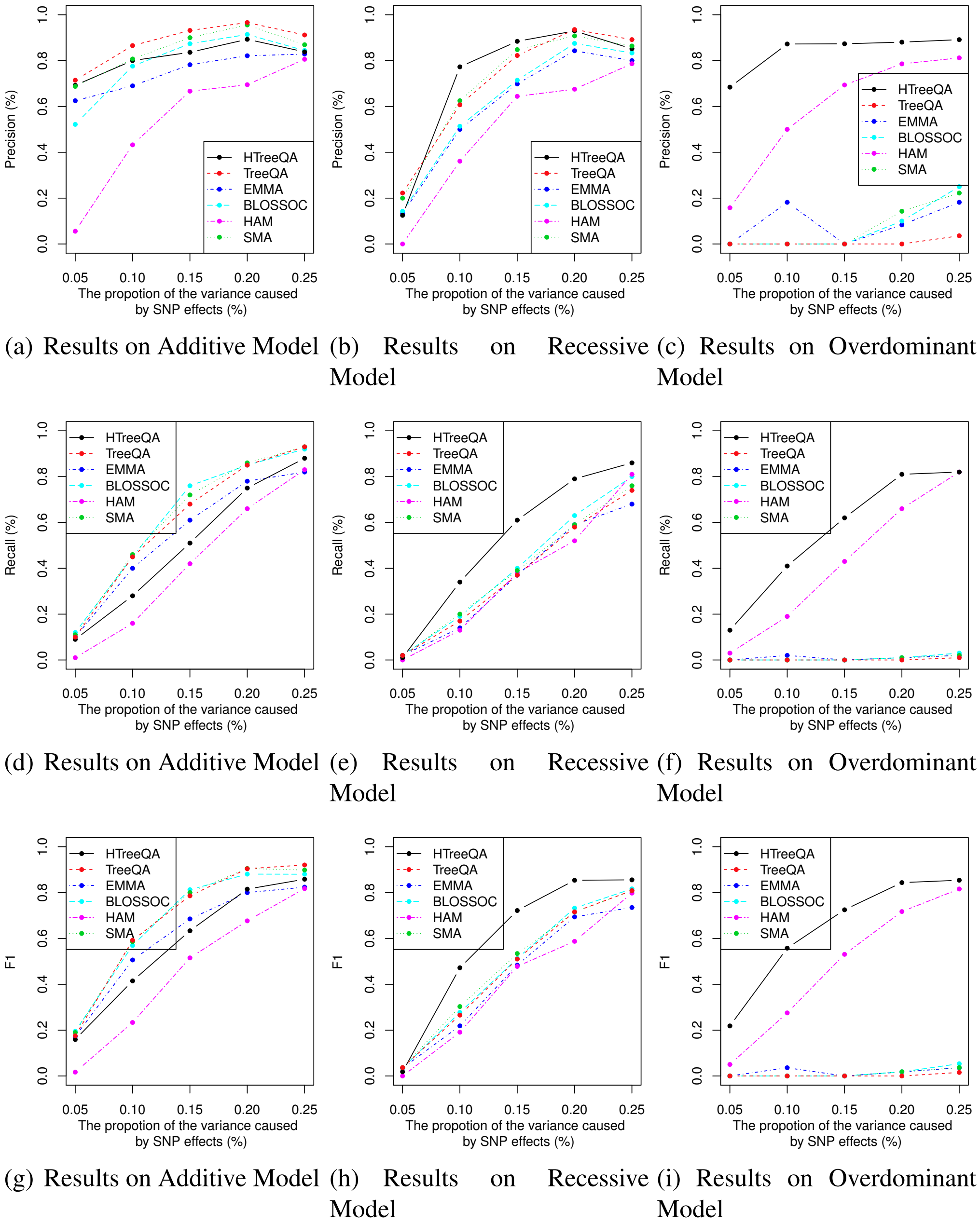
(f) Result of SMA

Figure 4.6: QTLs for mice daily average running distance. Only the SNPs that have top 0.5% -log(p-value) or BLOSSOC score are plotted in the figure. The dashed line is the significance level FWER = 0.05.

4.4.7 Simulation Study

To examine the performance of HTreeQA in a controlled environment, we simulate three different types of effects: additive, recessive and overdominant. For each selected method, only the SNPs with significance level FWER=0.05 are reported as QTLs. Since the causative SNPs are removed in the simulated data before the QTL analysis, in order to measure the accuracy of the result, a reported QTL is considered as a true positive when it is located within 50 SNPs from the causative SNP. Three measurements are measured to estimate the performance of each method: *precision*, *recall* and *F1 score*. Precision is defined as the ratio between the number of true QTLs that are detected and the total number of detected QTLs. Recall is defined as the ratio between the number of true QTLs that are detected and the total number of true QTLs that are simulated. The F1 score is the harmonic mean of precision rate and recall rate, and is defined as follows:

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.$$



(a) Results on Additive Model (b) Results on Recessive Model (c) Results on Overdominant Model

(d) Results on Additive Model (e) Results on Recessive Model (f) Results on Overdominant Model

(g) Results on Additive Model (h) Results on Recessive Model (i) Results on Overdominant Model

Figure 4.7: Comparison of HTreeQA, TreeQA, SSA, BLOSSOC, EMMA, and HAM under different genetic models.

Figure 4.7 shows the comparison of the selected methods. HTreeQA shows comparable performance to that of other methods in the additive model. In the recessive model and the overdominant model, HTreeQA demonstrates significant advantage over other methods. Since HTreeQA does not have any assumption of the type of genetic effect, it offers consistent power for detecting any effect. Other methods except HAM implicitly assume the additive model.

The phasing step required by the phylogeny-based methods BLOSSOC and TreeQA (for handling heterozygosity) will impair their ability in detecting associations between the phylogeny and the phenotype. The extent of its effect varies for different genetic models, especially with regard to heterozygous samples. It affects the additive model the least and overdominant model the most. For a homozygous sample the nodes corresponding to the two haplotypes carry the same allele, and thus their phenotypes always belong to the same allele group. This may cause minor inflation of the QTL signals since the two haplotypes are treated as independent samples by these methods. For a heterozygous sample the two haplotypes carry different alleles and therefore their corresponding nodes and phenotype are in two allele groups. Under the additive model assumption, one allele group contains all homozygous samples with high phenotype values and the other contains all homozygous samples with low phenotype values. The heterozygous samples have medium phenotype values which are added to both allele groups. This may cause minor deflation of the QTL signals. This is why all selected methods have comparable performance. TreeQA slightly outperforms others because its local phylogeny trees can well model the local population structure and separate QTL signals from genetic background.

However, under the assumption of overdominant model, heterozygous samples may have extreme phenotype values (beyond the range of phenotype values of the homozygous samples). These extreme phenotype values will always be in both allele groups; therefore, the phylogeny representation for phased data cannot explain the overdomi-

Methods	Running Time	Require Haplotype Reconstruction?
SMA	10 minutes	No
BLOSSOC	40 hours	Yes
HAM	20 minutes	No
TreeQA	40 hours	Yes
EMMA	3 hours 20 minutes	No
HTreeQA	12 minutes	No

Table 4.3: Running time comparison of the selected methods. The running time is measured on a machine with Intel i7 2.67GHz CPU and 8G memory.

nant effects at all. That is why the traditional phylogeny-based methods like BLOSSOC and TreeQA fail under such a model. Note that HTreeQA does not require phasing. The tri-state semi-perfect phylogeny tree has a partition that separates the heterozygous samples from the homozygous samples and thus it is able to detect an overdominant effect. Under the recessive model assumption, the heterozygous allele carries the same effect as one of the two homozygous alleles. Thus the impact of assigning haplotypes of the heterozygous samples to the two allele groups is greater than that under the additive model and is not as great as that under the overdominant model. Again, this does not affect HTreeQA. Overall, HTreeQA has the best performance in recessive models and overdominant models.

4.4.8 Running Time Comparison

The running time for each selected method are measured on a machine with Intel i7 2.67GHz CPU and 8G memory. All methods are tested using a dataset containing 180K SNPs and 184 individuals.

Table 4.3 shows the running time of these methods. If phasing is required, this step usually takes over 40 hours, and dominates the running time. HTreeQA demonstrates a huge advantage by completely avoiding haplotype reconstruction. It is over 600 times faster than the other methods that require haplotype data. HTreeQA is 15 times faster than EMMA, because HTreeQA does not need to explicitly incorporate the effect of global population structure as EMMA does. The running time of HTreeQA is comparable to that of SMA and HAM, the simplest models for QTL studies. They are not as effective as HTreeQA as demonstrated in the real phenotype and simulation studies.

4.4.9 The Choice between HTreeQA, TreeQA, and EMMA

HTreeQA can handle heterozygous genotype properly. It is suitable for genome wide association study on any multiparent crosses including the incipient CC lines, Heterogeneous Stock, and Diversity Outbred, as well as Recombinant Inbred Crosses (RIX) of CC lines. TreeQA is the best choice if one focuses on the additive effects. EMMA can correct for global population structure but is not able to address any local population structure. It degenerates to a simple linear model when applied to CC population with an evenly distributed global population structure as shown in Section 4.4.2. This represents a limitation of EMMA since local population structures exist in every mammalian resource, even though I only show the result in the Collaborative Cross population.

4.5 Conclusions

I propose a novel approach for local phylogeny-based QTL mapping on genotypes without haplotype reconstruction. I analyze the incipient Collaborative Cross, and show that there is no significant global population structure but visible local population structure. Such local population structure may bias the QTL mapping if it is not addressed properly. The notion of a tri-state semi-perfect phylogeny tree is introduced to represent accurate genetic relationships between samples in short genomic regions. As a generalization of the perfect phylogeny tree (defined on haplotypes), a tri-state semi-perfect phylogeny tree treats the heterozygous allele as the third state. It provides the power of modeling a wide range of genetic effects and delivers unbiased and consistent performance. This is a significant advantage over any previous methods that have strong bias towards an additive model. It is also worth noting that HTreeQA is much more computationally efficient than any alternative approach.

CHAPTER 5

DIPLOFFECT: BAYESIAN MODELING OF HAPLOTYPE EFFECTS IN MULTIPARENT POPULATIONS

In the previous Chapter, I presented HTreeQA for effectively discovering QTLs in multiparent line for both inbred and outbred populations. Discovering QTLs in multiparent crosses is just the initial step for understanding the underlying mechanism of complex traits, and the next important question is that how to estimate the effect sizes and the confident intervals for different genetic factors, aka, the founders genetic information, at the QTL. The statistically valid effect sizes and their confidence intervals are the foundations for interpreting how the QTL affects the quantitative trait and helping design and develop further biological experiments to validate the discovery. In this chapter, Diploffect is presented for estimating the effect sizes of different founders at the QTL.

5.1 Introduction

In this chapter, I also consider both types of populations, inbred or outbred, but make the assumption that, as with the examples in the previous chapters, organisms are diploid and founders are inbred. Because the genome of each individual in multiparent population can be described as a mosaic of founder haplotypes, any given locus in that genome can likewise be described in terms of the pair of haplotypes (ie, diplotype) present. In a panel of recombinant inbred lines with J founders (such as CC or MAGIC), a given locus can present with one of J homozygous diplotypes; the effect of each founder can be described by J (additive) haplotype effects. In an outbred population with J founders (such as the HS or DO), $J(J+1)/2$ distinct diplotypes are possible (or J^2 if distinguishing parent of origin); characterizing a QTL in these populations means accounting all $J(J+1)/2$ (or J^2) diplotype effects. The identity of the presenting diplotype in each case cannot be observed directly but can be probabilistically inferred from genotype data. A number of algorithms have been developed to do this, notably those based on a hidden Markov model (HMM) formulation (eg, HAPPY, Mott et al. (2000); GAIN Liu et al. (2010); although see also Bauman et al. (2008)). In the HMM framework, diplotypes are modeled as latent outcomes drawn from a discrete set of possibilities; genotype data provides partial information about this underlying latent state, and so the HMM's reconstruction of the haplotype mosaic leads to haplotype assignments that are probabilistic — for each individual at each locus, a list of inferred probabilities for each possible diplotype.

When mapping QTL in multiparent populations, testing for genetic association using this inferred probabilities of diplotypes rather than using the observed alleles at genotyped markers confers several advantages. First, all ungenotyped genetic variants are modeled automatically; this includes not only ascertained SNPs but also all the SNPs they fail to tag. Second, HMM-based haplotype inference helps guard against genotyping or sequencing error while providing robust imputation for cases where genotyping has failed. Third, the inferred probabilities provide much more informative picture of underlying genetic factors than the genotype data, resulting higher mapping resolution than using observed genotype data.

Strictly speaking, genetic association with inferred haplotypes is most properly handled through some form of mixture model: At a given locus, the QTL effects are estimated conditional on haplotype composition; haplotype composition is itself modeled probabilistically as described previously; and the resulting likelihood, which includes alternative haplotype configurations, is used for significance testing (Lander and Botstein, 1989). Mixture models are, however, computationally demanding to a degree that usually makes them impractical for large scale association. A fast and powerful alternative, advanced by Haley and Knott (1992) for 2-parent crosses and later Mott et al. (2000) for 8-parent crosses, is to use regression: Treat the inferred haplotype probabilities as if they were observed haplotype "dosages", include these dosages as a feature vector of primary factors affecting the phenotype in a linear model, then perform an ANOVA-like test for a significant effect of that feature vector. This approach, which I refer to as Regression on Probabilities (ROP), is not only highly scalable but also, due to the ubiquity and flexibility of linear modeling software, is relatively simple to implement, at least once haplotype probabilities are provided. Indeed, for LD-based genetic association in multiparent populations, ROP has become the dominant approach (Aylor et al., 2011b; Valdar et al., 2006; Svenson et al., 2012; Kover et al., 2009).

Despite ROP's power in detecting QTL, however, when it comes to subsequent characterization of QTL effects has the ROP approach has severe shortcomings. Problems arise because ROP is a linear non-hierarchical approximation to a hierarchical mixture model: The numerical output of the HMM is treated as if it were an arbitrarily scaled feature matrix rather than a probabilistic description of a categorical state. The extent to which this matters depends on both the degree and type of uncertainty present. In the best case, when haplotype assignment is certain for all individuals, and the probabilities therefore reduce to a design matrix of ones and zeros, ROP produces valid inference. In the presence of uncertainty, however, several complications follow. The inability of haplotype reconstruction to distinguish diplotypes at loci where some founders are identical (by state or descent) can produce a design matrix that is multicollinear, causing the model to become non-identifiable. Although this non-identifiability can be circumvented for the purposes of fitting a predictive model, it is at the cost of downstream interpretability. For example: applying a full rank factorization to the matrix (as in, eg, Appendix A of Valdar et al. (2009)) produces a model that can be fitted but with estimated parameters that are uninterpretable; applying a ridge-type penalty (as in, eg, (Woods et al., 2012)) preserves identifiability but only artificially — introducing an arbitrary parameter whose presence invalidates deeper inference (eg, confidence intervals). Even when multicollinearity is mild enough for all effects to remain identifiable, uncertainty still produces an uneven narrowing of the the numerical range of the dosages, which in turn lead to ROP estimates of effects becoming inflated in complicated ways (Broman and Sen, 2009; Ronnegard and Valdar, 2011a).

Ideally then, estimation of QTL effects in a multiparent population should: 1) incorporate the probabilistic information from haplotype reconstruction; 2) accommodate not only additive haplotype effects but also (at least) the effects of dominance, as would occur in outbred or incompletely inbred populations; 3) use shrinkage to moderate imbalanced and sparsely sampled representation of a potentially large number diplotypes; 4) be flexible enough to incorporate confounding sources of variation such as polygenic effects, complex effects of batching, and so on. To the best of our knowledge, no existing method fulfills all of these criteria. A few, however, address at least the first — appropriate handling of haplotype uncertainty. Sillanpaa and Arjas (1998, 1999) used a Bayesian approach when parental information is missing in inbred and outbred population respectively, but are not able to incorporate prior haplotype probabilities from haplotype reconstruction and handle only biallelic data. Kover et al. (2009) applied a multiple imputation approach by sampling the unobserved haplotypes from the inferred haplotype matrices and averaging standard regression the least square estimates on imputed datasets. Durrant and Mott (2010) developed a partially Bayesian mixed model of QTL mapping based on inferred haplotypes that mostly satisfies (1) and (3) above. However, their prior for the haplotype effects in the model is very rigid to reach a complete factorization of the likelihood, and this restricts application of their model to phenotypes that are normally distributed and unaffected by polygenic effects.

Here I describe a flexible statistical model, Diploffect (Zhang et al., 2014), for estimating effects of haplotypes and diplotypes at QTL detected in multiparent populations. Using a Bayesian hierarchy that induces variable shrinkage, Diploffect obtains full posterior distributions for additive and dominance effects that take account of both uncertainty in the haplotype composition at the QTL and confounding factors such as polygenic or family effects. In basing Diploffect model around existing, extendible software, I describe a flexible framework that accommodates non-normal phenotypes. In addition, by using a model that is fully Bayesian, Diploffect exploits an opportunity untouched by earlier methods: The potential, when phenotypes and uncertain haplotypes are modeled jointly, for phenotypic data to inform and improve inference about haplotype configuration at the QTL as well as vice versa (see, for example, a related application in Lin and Zeng (2006)). To provide practical solutions and perspectives about relative trade-offs, I demonstrate two implementations of Diploffect, and compare their performance in terms of accuracy and running time to simpler procedures.

5.2 Statistical Models and Methods

The approach is fully Bayesian: I advance a framework that models latent parameters as outcomes of higher order processes and leads to coherent inference and prediction given observed data and prior uncertainty. To provide flexibility in both the form of inference and the likelihood assumptions of the phenotype, I developed two different approaches to estimate the posterior distributions: Markov Chain Monte Carlo (MCMC) sampling and Importance Sampling (IS). First, I describe a decomposition of QTL effects based on haplotypes, known or uncertain, in general, and one way in which Bayesian inference of those effects can naturally proceed. Then I describe the model and the computation methods used to fit it. Last, I describe for comparison several non-Bayesian regression-based approaches to haplotype effect estimation, relating them back to the original framework.

5.2.1 Haplotypes and Diplotype States

In a multiparent population comprising individuals $i = 1, \ldots, n$ descended from a smaller set of diploid founders $j = 1, \ldots, J$, the genetic state at each locus in each individual can be described in terms the pair of founder haplotypes (ie, the diplotype) present — that is, in terms of the diplotype state. I encode the diplotype state for individual i at locus m using a $J \times J$ indicator matrix $\mathbf{D}_i(m)$, where for maternally inherited founder haplotype $j \in \{1, \ldots, J\}$ and paternally inherited haplotype $k \in \{1, \ldots, J\}$, corresponding to diplotype jk, the entry in the jth row and kth column is $\{\mathbf{D}_i(m)\}_{jk} = 1$ and all other elements are zero. A diplotype is defined as homozygous when $j = k$, and heterozygous when $j \neq k$; under the heterozygote diplotype, when parent of origin is unknown or disregarded, $jk \equiv kj$ and it is assumed that $\{\mathbf{D}_i(m)\}_{jk} + \{\mathbf{D}_i(m)\}_{kj} = 1$.

5.2.2 Haplotype Effects at a QTL

Given a a trait of interest observed on the n individuals, $\mathbf{y} = y_1, \ldots, y_n$, the effect of substituting one diplotype for another on that trait's value can be expressed using a generalized linear model of the form $y_i \sim \text{Target}(\text{Link}^{-1}(\eta_i), \xi)$, where Target is the sampling distribution, Link is the link function, η_i is a predictor whose value depends on diplotype state (and other modeled properties of the individual) and which acts through the link function to adjust the expected value of y_i, and ξ represents other parameters in the sampling distribution; for example, with a normal target distribution and identity link, $y_i \sim \text{N}(\eta_i, \sigma^2)$, and $\text{E}(y_i) = \eta_i$.

Under the assumption that haplotype effects combine additively to influence the phenotype, the linear predictor can be written as

$$\eta_i = \mu + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{add}(\mathbf{D}_i(m)) \tag{5.1}$$

where $\mathbf{add}(\mathbf{X}) = \mathbf{1}^{\mathsf{T}}(\mathbf{X}+\mathbf{X}^{\mathsf{T}})$ such that $\boldsymbol{\beta}$ is a J-vector of (additive) *haplotype effects*, and μ is an intercept term that, in expectation, makes $\boldsymbol{\beta}$ sum to zero. The assumption of additivity can be relaxed to admit effects of dominance by introducing a *dominance deviation*:

$$\eta_i = \mu + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{add}(\mathbf{D}_i(m)) + \boldsymbol{\gamma}^{\mathsf{T}}\mathbf{dom}(\mathbf{D}_i(m))\,, \tag{5.2}$$

where the appropriate definition of $\mathbf{dom}(\mathbf{X})$, and therefore $\boldsymbol{\gamma}$, depends on whether the effects of reciprocal heterozygous diplotypes jk and kj are modeled to be equivalent. If they are, then dominance can be modeled as *symmetric*: $\mathbf{dom}(\mathbf{X})$ is defined as $\mathbf{dom.sym}(\mathbf{X}) = \text{vec}(\text{upper.tri}(\mathbf{X} + \mathbf{X}^{\mathsf{T}}))$, where upper.tri() returns only elements above the diagonal of a matrix, and effects vector $\boldsymbol{\gamma}$ is length $(J^2 - J)/2$. Otherwise, if diplotype effects are modeled to differ by parent-of-origin, then dominance is *asymmetric*: $\mathbf{dom}(\mathbf{X})$ is defined as $\mathbf{dom.asym}(\mathbf{X}) = \text{vec}(\text{off-diag}(\mathbf{X}))$, where off-diag returns all off-diagonal elements, and $\boldsymbol{\gamma}$ is length $J^2 - J$. Throughout the remainder of the paper, for simplicity, dominance will be modeled as symmetric. Lastly, for notational convenience, I define the *diplotype effects*, $\boldsymbol{\delta}$, as

$$\delta_{jk} = \beta_j + \beta_k + I(j \neq k)\gamma_{(jk)}\,, \tag{5.3}$$

for all distinguishable jk.

5.2.3 Haplotype Inference and Diplotype Probabilities

In practice, the diplotype state at a locus m cannot be observed directly, but it can be inferred probabilistically from genotype data. Denoting available genotype data on individuals as $\mathcal{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$, and genotype information on the founders as

$\mathcal{H} = \{\mathcal{H}_1, \ldots, \mathcal{H}_J\}$, haplotype reconstruction algorithms based on a hidden Markov model typically seek to estimate for each individual i at each locus $m = 1, \ldots, M$ a $J \times J$ matrix of inferred diplotype probabilities,

$$\mathbf{P}_i(m) = p(\mathbf{D}_i(m)|\mathcal{G}_i, \mathcal{H})\,, \tag{5.4}$$

where each element $\{\mathbf{P}_i(m)\}_{jk}$ contains the probability that diplotype jk is present, and where in more sophisticated algorithms additional terms may be present in the conditioning statement (eg, $\mathcal{G}$ in place of $\mathcal{G}_i$). Diplotype state is therefore modeled as if drawn from a categorical distribution with probability parameter $\mathbf{P}_i(m)$, ie,

$$\mathbf{D}_i(m) \sim \text{Cat}(\mathbf{P}_i(m))\,, i = 1, \ldots, n\,. \tag{5.5}$$

In the HAPPY formulation (Mott et al., 2000), which I adopt here, element $\{\mathbf{P}_i(m)\}_{jk}$ is the HMM-derived Baum-Welsh probability of diplotype jk, averaged over the interval between two adjacent markers m and $m+1$. In other words, $\{\mathbf{P}_i(m)\}_{jk}$ is approximately the probability that a randomly chosen point within the interval inherits from the diplotype jk. When descent is unambiguous, $\mathbf{P}_i(m) = \mathbf{D}_i(m)$; otherwise $\mathbf{P}_i(m)$ represents a hedged bet on which diplotype occurs in the interval, and typically becomes less informed as a function of marker sparsity, recombination density, and genotyping error.

5.2.4 Regression On Probabilities

When diplotype state is available only probabilistically, rather than known as in Eq 5.1 and 5.2, accurate modeling of haplotype effects at the QTL is more challenging: Inference, if it is to be accurate, must now take into account not only variability of estimates due to sampling but also variability due to uncertainty in the predictor values themselves (see later). For purposes other than haplotype estimation, however, it often suffices to use the simple approximation of Regression on Probabilities

	True Diplotype Assignment		Inferred Diplotype Probability		
Individual	A	B	A	B	Phenotype
1	1	0	0.51	0.49	1
2	0	1	0.49	0.51	0

Table 5.1: Illustrative example of true diplotype state vs inferred diplotype probabilities for two individuals at one genetic locus.

(introduced in the biallelic context by Haley and Knott (1992)). In ROP, matrices describing diplotype state $\mathbf{D}_1(m), \ldots, \mathbf{D}_n$ are replaced by those of diplotype probabilities $\mathbf{P}_1(m), \ldots, \mathbf{P}_n(m)$, and the additive model in Eq 5.1 is approximated as

$$\eta_i = \mu + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{add}(\mathbf{P}_i(m))\,, \tag{5.6}$$

with the model including dominance effects defined similarly as

$$\eta_i = \mu + \boldsymbol{\beta}^{\mathsf{T}}\mathbf{add}(\mathbf{P}_i(m)) + \boldsymbol{\gamma}^{\mathsf{T}}\mathbf{dom}(\mathbf{P}_i(m))\,, \tag{5.7}$$

In these approximations, the predictors $\mathbf{add}(\mathbf{P}_i(m))$ and $\mathbf{dom}(\mathbf{P}_i(m))$ are treated as arbitrary-valued feature vectors while $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are estimated as their best-fitting coefficients. This formulation is powerful for regression-based significance testing; for obtaining meaningful substitution effects, however, it is problematic.

An artificial example is given in Table 5.1, in which at a given QTL each of two individuals have one of two diplotypes, A or B. Regression on known diplotypes estimates the substitution effect as 1; regression on diplotype probabilities, which in this example are highly uncertain but nonetheless accurate in the sense of placing more probability on the right answer, estimates the effect as 50.

The following R code provides the script for estimating the effect:

```
x = matrix(c(0.51, 0.49, 0.49, 0.51), nrow = 2)
y = c(1,0)
lm(y ~ x)
```

Both estimates fit their input data equally well; applied to new inputs of the *same form* (specifically, the same degree of uncertainty), they would give equally accurate predictions. If, however, the ROP estimate of 50 was used to predict phenotype for individuals where diplotype is known (or even where it is inferred with greater certainty than in Table 5.1), poor accuracy would clearly result.

As the number of possible diplotype states grows, the problem of inflated estimates increases and is compounded with additional problems of multicollinearity, whereby higher order confounding in diplotype inference leads to linear dependence that in turn reduces the effective number of estimable parameters in $\boldsymbol{\beta}$ (see, eg, Appendix A of Valdar et al. (2009)). It is often intuitively appealing to regard $\mathbf{add}(\mathbf{P}_i(m))$ as a set of "haplotype dosages"; however, even without multicollinearity, the fact that the degree of uncertainty in $\mathbf{P}_i(m)$ will differ among individuals means that uncertainty and dosage are confounded, and the corresponding $\boldsymbol{\beta}$ estimated by ROP does not truly estimate a "haplotype dosage effect".

5.2.5 Diploffect Model

Estimating haplotype effects in a way that incorporates uncertainty in the diplotype state requires a full probabilistic model. Here I describe one such model, Diploffect, which estimates haplotype effects (and related parameters) contingent on diplotype state while simultaneously modeling diplotype state itself as an unknown parameter whose distribution is informed *a priori* by probabilities from an HMM-based haplotype reconstruction algorithm.

Diploffect uses a Bayesian framework in which diplotypes $\boldsymbol{\Delta} = \{\mathbf{D}_1(m), \cdots, \mathbf{D}_n(m)\}$ and all effects $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \boldsymbol{\gamma}, \dots\}$ (ie, all non-diplotype parameters) are latent variables modeled in joint posterior distribution

$$p(\boldsymbol{\theta}, \boldsymbol{\Delta}|\boldsymbol{\Psi}, \mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\Delta})p(\boldsymbol{\Delta}|\boldsymbol{\Psi})p(\boldsymbol{\theta})\,, \tag{5.8}$$

which conditions on inferred probabilities from the haplotype reconstruction $\boldsymbol{\Psi} = \mathbf{P}_i(m), \cdots, \mathbf{P}_n(m)$ and observed phenotype data $\mathbf{y}$. In this specification, the phenotype is modeled in the likelihood $p(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\Delta})$ as a function of the effects $\boldsymbol{\theta}$ and the diplotypes $\boldsymbol{\Delta}$, described in more detail below; the diplotypes $\boldsymbol{\Delta}$ are modeled as latent categorical variables with prior $p(\boldsymbol{\Delta}|\boldsymbol{\Psi})$. This has two important consequences. First, the posterior distribution of effects

$$p(\boldsymbol{\theta}|\boldsymbol{\Psi}, \mathbf{y}) = \int p(\boldsymbol{\theta}, \boldsymbol{\Delta}|\boldsymbol{\Psi}, \mathbf{y})\mathrm{d}\boldsymbol{\Delta}\,, \tag{5.9}$$

from which all estimates and intervals of haplotype effects can be obtained, averages over plausible diplotype configurations; this leads to effect estimates of $\boldsymbol{\theta}$, including interval estimates, that incorporate uncertainty in diplotype state. Second, a posterior distribution is generated for the diplotype state $\boldsymbol{\Delta}$ conditional on the phenotype $\mathbf{y}$:

$$p(\boldsymbol{\Delta}|\boldsymbol{\Psi}, \mathbf{y}) = \int p(\boldsymbol{\theta}, \boldsymbol{\Delta}|\boldsymbol{\Psi}, \mathbf{y})\mathrm{d}\boldsymbol{\theta}\,. \tag{5.10}$$

This posterior is a Bayesian update of prior $p(\boldsymbol{\Delta}|\mathcal{G}, \mathcal{H}) = \boldsymbol{\Psi}$ (see Eq 5.4) in light of phenotypic information. Specifically, since the prior of diplotypes is a categorical distribution, the marginal posterior of the diplotype is also categorical:

$$p(\mathbf{D}_i(m)|\Theta, \boldsymbol{\Psi}, \mathbf{y}) \sim \mathrm{Cat}(Q(\mathbf{D}_i(m)_{11}), Q(\mathbf{D}_i(m)_{12}), ..., Q(\mathbf{D}_i(m)_{JJ}). \tag{5.11}$$

where

$$Q(\mathbf{D}_i(m)_{jk}) = p(\mathbf{D}_i(m)_{jk} = 1|y_i, \boldsymbol{\theta}, \mathbf{P}_i(m)) \propto \underbrace{\mathbf{P}_i(m)_{jk}}_{\text{prior}} \times \underbrace{p(y_i|\boldsymbol{\theta}, \mathbf{D}_i(m)_{jk} = 1)}_{\text{likelihood}}\,. \tag{5.12}$$

This reflects the following intuition: Suppose prior to observing $\mathbf{y}$, diplotype probabilities $\mathbf{P}_1, \ldots, \mathbf{P}_{n-1}$ are well informed but $\mathbf{P}_n$ is not; if analysis with $\mathbf{y}$ reveals a clear pattern of effects (eg, high phenotypes associated with particular diplotype states) then y_n provides information to update $\mathbf{P}_n$. Moreover, it implies that different phenotypes could in theory promote different underlying diplotype states $\boldsymbol{\Delta}$ — a particularly useful feature when locus m is defined broadly enough to contain multiple recombinants and therefore multiple configurations of $\boldsymbol{\Delta}$ of which only one is relevant to the QTL.

Because the likelihood $p(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{\Delta})$ conditions on diplotypes $\boldsymbol{\Delta}$, haplotype effects can be modeled in the linear predictor relative to diplotype state (as in 5.2) rather than to diplotype probabilities (as in 5.6). The linear predictor for individual i is modeled as

$$\eta_i = \mu + \underbrace{\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{x}_i}_{\substack{\text{covariates} \\ (optional)}} + \underbrace{\boldsymbol{\beta}^{\mathsf{T}}\mathbf{add}(\mathbf{D}_i(m))}_{\text{additive haplotype effects}} + \underbrace{\boldsymbol{\gamma}^{\mathsf{T}}\mathbf{dom}(\mathbf{D}_i(m))}_{\text{dominance deviation}} + \underbrace{u_{r[i]}}_{\substack{\text{family/polygenic} \\ (optional)}} . \tag{5.13}$$

Additional parameters are included for covariates and genetic background: $\boldsymbol{\alpha}$ models effects of covariates in $\mathbf{x}_i$; and $u_{r[i]}$ models, for example, the effect of sibship (or CC line, Aribidopsis cousin line, etc) $r[i]$ to which to individual i belongs, and $u_r \sim \mathrm{N}(0, \tau_u^2)$ — alternatively, where computationally feasible, $u_r[i]$ is a polygenic effect after, eg, Kennedy et al. (1992); Cheng et al. (2011). These additional effects are loosely specified: I present the model within established, extensible software that allows users to define more complex structured effects easily.

The haplotype effects $\boldsymbol{\beta}$ and dominance deviations $\boldsymbol{\gamma}$ are modeled hierarchically, as if drawn from multivariate normal distributions $\boldsymbol{\beta} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}\tau^2_{\text{add}})$ and $\boldsymbol{\gamma} \sim \mathrm{N}(\mathbf{0}, \mathbf{I}\tau^2_{\text{dom}})$. Hierarchical modeling in this case not only reflects the fact that diplotype effects are expected to lie on similar scales and should therefore inform each other but also that inference of effects represents a decision problem involving the estimation of many parameters simultaneously and so naturally benefits from hierarchical shrinkage (Parmigiani and Inoue, 2009; Gelman and Hill, 2007). Shrinkage is particularly important here because many of the diplotypes will be sparsely sampled, with some missing entirely; in the face of this, hierarchical shrinkage leads to posteriors that are stable but vague rather than unstable and erratic.

The remaining parameters are given vague, conjugate priors: $(\mu, \boldsymbol{\alpha}) \sim \mathrm{N}(\mathbf{0}, \mathbf{I}c)$, where c is large relative to the phenotype scale (eg, $c = 1000$ for $\mathrm{Var}(y) = 1$); τ^2_{add}, τ^2_{dom}, and τ^2_{u} are given inverse-gamma priors as in, eg, Lenarcic et al. (2012). The complete Diploffect model is summarized using plate notation in Figure 5.1.

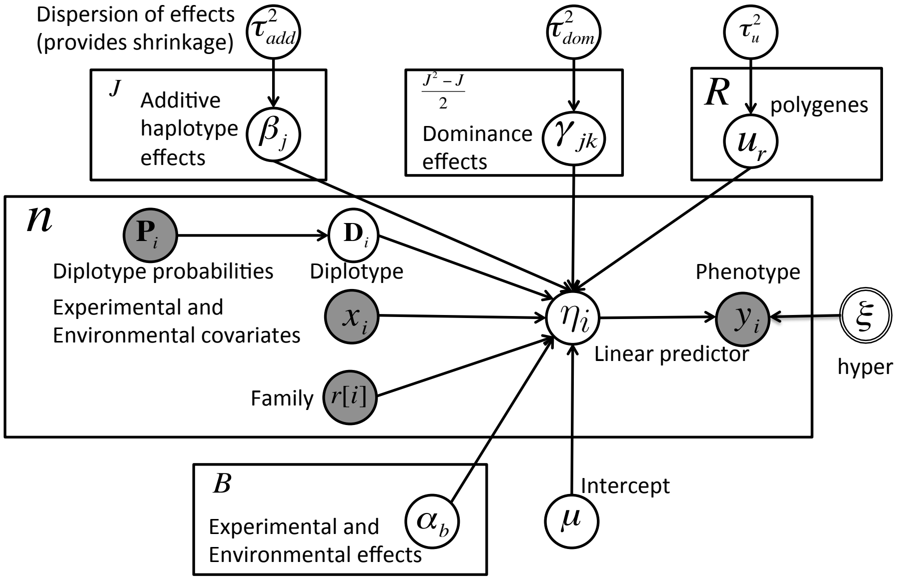


Figure 5.1: The plate notation for Diploffect model with dominance deviation effects and kinship effects. The priors of the effects are omitted in the plate notation. The nodes with grey background represent the observed data, the nodes with white background and a single circle represent the unknown variables, and the nodes with double circles represent the remaining parameter except the linear predictor in the generalized linear model.

5.2.6 Diploffect Estimation by MCMC: `DF.MCMC`

Posteriors for the parameters of the Diploffect model can be estimated by Markov Chain Monte Carlo through iteration of two basic steps:

1. Sample all effect variables $\boldsymbol{\theta}^{(k+1)}$ given the previous iteration's diplotypes $\boldsymbol{\Delta}^{(k)}$:

$$\boldsymbol{\theta}^{(k+1)} \sim p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\Delta}^{(k)}) \tag{5.14}$$

2. Sample diplotypes $\boldsymbol{\Delta}^{(k+1)}$ given effect variables $\boldsymbol{\theta}^{(k+1)}$:

$$\boldsymbol{\Delta}^{(k+1)} \sim p(\boldsymbol{\Delta}^{(k+1)}|\boldsymbol{\Psi}, \boldsymbol{\theta}^{(k+1)}, \mathbf{y}) \tag{5.15}$$

where k is the index of the iteration and K is the total number of iterations (for k = 1, ..., K). Initial values for $k = 1$ are randomly sampled from their priors. Step 1 is relatively straightforward because given $\boldsymbol{\Delta}$, the remaining forms a generalized linear model (GLM) whose efficient computation is well studied (Plummer, 2003). Step 2, however, requires special consideration.

A straightforward approach for step 2 is to evaluate all diplotypes' posterior probabilities in $\mathbf{D}_i(m)$ by Eq 5.12, and draw one individual's diplotype at a time from the posterior categorical distribution. This requires $O(J^2)$ computational time per individual because it requires evaluating the function Q for all diplotypes. For the sake of efficiency, I develop a new method, Discrete Slice Sampling with Prior Reordering, described in the following section, which significantly reduces this computational time; throughout the paper, I will refer to this method in short form as `DF.MCMC`. The proof of the convergence of this approach, which embeds slice samplers in a MCMC chain, can be found at Neal (2003) (it proves that using univariate variables in a MCMC chain does not change the invariant distribution of the chain).

Discrete Slice Sampling with Reordering of Prior Probabilities

To help efficiently traverse the space of possible diplotype states, we propose an optimization of the discrete slice sampling algorithm described by Neal (2003). This optimization begins by reordering all $J \times (J + 1)/2$ entries in matrix $\mathbf{D}_i$. Let $T(jk)$

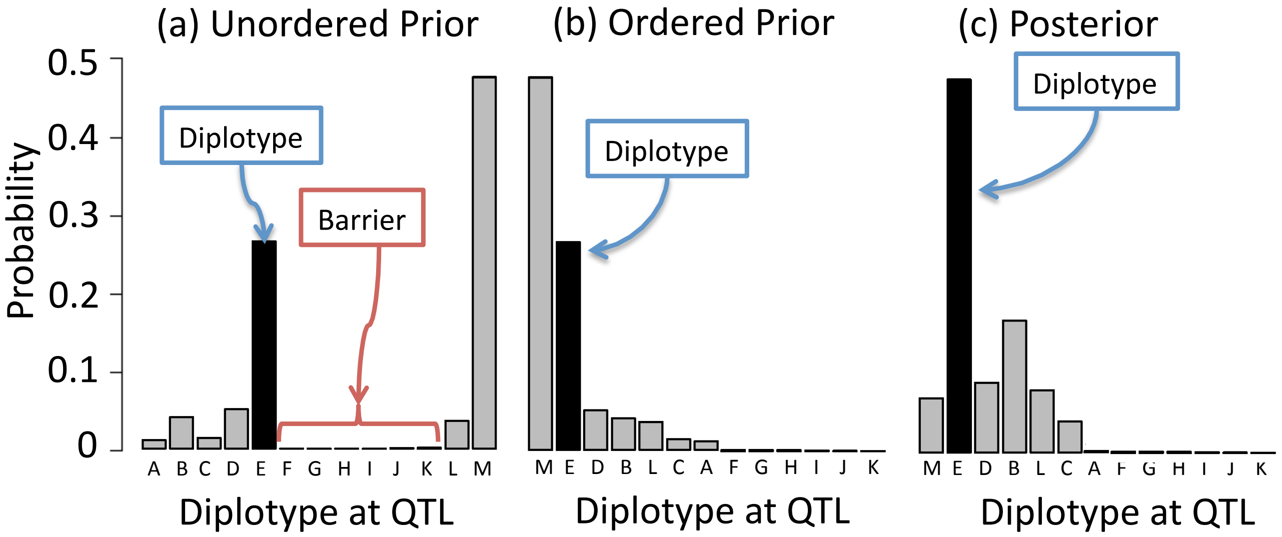


Figure 5.2: Reordering of prior probabilities in the discrete slice sampler, using as an example the diplotype probabilities from haplotype reconstruction (using HAPPY) on the Pre-CC. Diplotypes are represented by different letters, and 23 diplotypes with very low probabilities are omitted. The true diplotype, selected during simulation, is shaded black. The original ordering of diplotypes (from the HAPPY) is shown in (a), and illustrates the problem to be addressed: If the initially sampled diplotype is M, the slice sampler cannot easily cross the barrier region to sample other high probability diplotypes. Reordering the diplotypes by their prior probabilities to create a smoother distribution, as in (b), removes this barrier region, and allowing the sampler to move easily between its initial value and all other values of high to moderate probability. Panel (c) shows the posterior of this distribution given phenotype data (from the `DF.MCMC` procedure), in which the true diplotype's posterior probability is increased.

represent diplotype jk's order in the range $1, \ldots, J \times (J+1)/2$, and define two boundary diplotypes for $T(L) = 0$ and $T(R) = J \times (J+1)/2 + 1$ and set their posterior probabilities to zero. Therefore, for the previous diplotype x', we first evaluate $S = Q(\mathbf{D}_i(m)_{T^{-1}(x')})$, then sample an auxiliary variable $q \sim U(0, S)$. We expand a region $[l, u]$ satisfying $Q(l) \geq q$ and $Q(l - l) < q$ and $Q(u) < q$ and $Q(u) \geq q$. From uniform distribution defined on $[u, l]$, we keep sampling the new diplotype status x^{new} until we reach one for which $Q(\mathbf{D}_i(m)_{T^{-1}(x)}) \geq q$. The diplotype posterior in Eq 5.12 is only evaluated a few times during each iteration, thus it is much faster than full sampling.

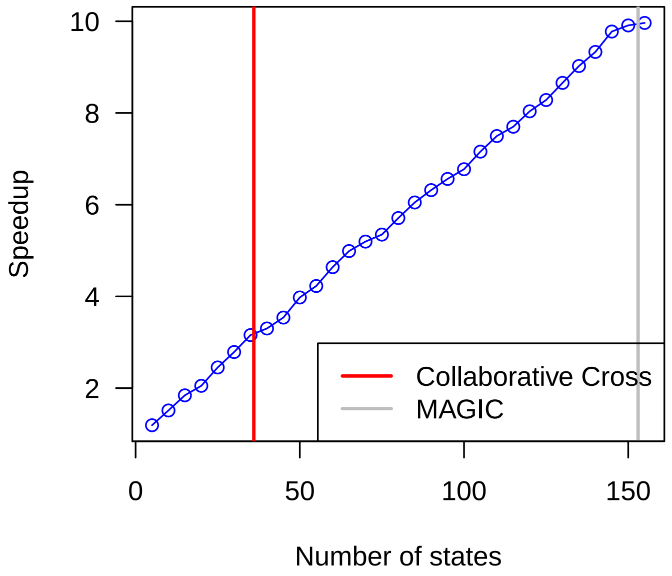


Figure 5.3: The speed up of using the discrete slice sampling with reordering instead of using draws from posterior multinomial distributions

If we directly apply discrete slice sampling with an arbitrary chosen order (e.g. alphabetical), the posterior samples can remain stuck within islands of higher scoring diplotypes flanked by lower-scoring ones, such that a long Markov Chain is required for adequate mixing, eg, Figure 5.2 (a). In order to avoid poor-mixing problem for sampling haplotypes and increase the efficiency of MCMC, we use a modified discrete sampling approach by reordering the prior categorical probabilities by the prior haplotype probabilities. In Figure 5.2 (b), the barrier is removed, and the posteriors in 5.2 (c) are mixing. The underlying true diplotype's posterior probability is much higher than the incorrect ones because of the joint estimation model on both haplotypes and their effects.

I compared this approach with the usual approach — to sample the posteriors by drawing from posterior multinomial distributions. Figure 5.3 shows how much speed up using the discrete slice sampling with reordering under various number of states (diplotypes): 3 times speed up for CC, and 10 times speed up for MAGIC. This indicates that this method can efficiently improve the running time of the MCMC approach.

5.2.7 Diploffect Estimation by Importance Sampling: `DF.IS` and `DF.IS.kinship`

Because calculation of posteriors from `DF.MCMC` requires some level of expertise in Bayesian computation, namely that of monitoring convergence of an iterating MCMC chain, I also provide a non-iterative strategy based on Importance Sampling (IS) of Integrated Nested Laplace Approximations (INLA). INLA provides a deterministic estimate of the multivariate posterior distribution of a GLM (Rue et al., 2009). In IS procedure, these GLM posteriors are estimated conditional on diplotype for many possible diplotype configurations; they are combined through reweighting to give a final mixture distribution that resembles more closely the integration of the full posterior in Eq 5.9. Specifically, in five steps, the procedure is:

1. Sample diplotypes $\boldsymbol{\Delta}^{(k)}$ from their prior

$$\boldsymbol{\Delta}^{(k)} \sim p(\boldsymbol{\Psi}), \tag{5.16}$$

2. Obtain an INLA estimate of posterior $p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\Delta}^{(k)})$ for effect variables $\boldsymbol{\theta}^{(k)}$.
3. Obtain an INLA estimate of the marginal likelihood $w^{(k)} = p(\mathbf{y}|\boldsymbol{\Delta}^{(k)})$.
4. Repeat steps 1 to 3 K times.
5. Estimate the posterior of any statistic of interest $T(\boldsymbol{\theta})$ using the weighted mixture

$$\widehat{T}_{\text{IS}}(\boldsymbol{\theta}) = \sum_k T(\boldsymbol{\theta}^{(k)}) w^{(k)} / \sum_k w^{(k)}\,, \tag{5.17}$$

where for each k, statistic $T(\boldsymbol{\theta}^{(k)})$ is calculated from the corresponding posterior $p(\boldsymbol{\theta}|\mathbf{y}, \boldsymbol{\Delta}^{(k)})$ calculated in step 2.

In the above, $T(\boldsymbol{\theta})$ could be anything from, for example, the posterior mean of $\boldsymbol{\beta}$ to the full density of $\boldsymbol{\theta}$. The calculation of the weighting function $w^{(1)}, \ldots, w^{(\mathrm{K})}$ uses the marginal likelihood obtained from INLA as I explain below.

In order to explain why the weight is the marginal likelihood of the model, I use a simplified model here for the proof. Assuming that we have two blocks of random variables $\mathbf{a}$ and $\mathbf{b}$, and the data is $\mathbf{y}$. The joint likelihood of the complete model is $p(\mathbf{y}|\mathbf{a}, \mathbf{b})p(\mathbf{a}, \mathbf{b})$. The prior of $\mathbf{a}$ and $\mathbf{b}$ are independent, which means $p(\mathbf{a}, \mathbf{b}) = p(\mathbf{a})p(\mathbf{b})$.

It might be hard to use MCMC sampling to jointly draw $\mathbf{a}, \mathbf{b}$ from their posteriors, for example, $p(\mathbf{a}|\mathbf{y}, \mathbf{b})$ is hard or costful to sample. Instead, we could use the following approach to sample from an importance density function $g(\mathbf{a}, \mathbf{b})$, and re-weight all samples by $w(\mathbf{a}, \mathbf{b}) = \frac{p(\mathbf{y}|\mathbf{a},\mathbf{b})p(\mathbf{a},\mathbf{b})}{g(\mathbf{a},\mathbf{b})}$.

For a limited number of iteration K, we repeat the following steps,

- **Step 1:** $\mathbf{a}_{IS}^{(k)} \sim p(\mathbf{a})$
- **Step 2:** $\mathbf{b}_{IS}^{(k)} \sim p(\mathbf{b}|\mathbf{y}, \mathbf{a}_{IS}^{(k)})$

where $\mathbf{a}_{IS}^{(k)}, \mathbf{b}_{IS}^{(k)}$ are the samples for the kth iteration. This approach is best if $p(\mathbf{b}|\mathbf{y}, \mathbf{a}_{IS}^{(k)})$ is relatively easy to calculated. The target density of $\mathbf{a}_{IS}^{(k)}, \mathbf{b}_{IS}^{(k)}$, which is also the importance density function, is $p(\mathbf{a})p(\mathbf{b}|\mathbf{y}, \mathbf{a})$.

Therefore,

$$w(\mathbf{a},\mathbf{b}) = \frac{p(\mathbf{y}|\mathbf{a},\mathbf{b})p(\mathbf{a},\mathbf{b})}{g(\mathbf{a},\mathbf{b})} \tag{5.18}$$

$$\propto \frac{p(\mathbf{a},\mathbf{b}|\mathbf{y})}{p(\mathbf{a})p(\mathbf{b}|\mathbf{y},\mathbf{a})} = \frac{p(\mathbf{a},\mathbf{b}|\mathbf{y})}{p(\mathbf{a})\frac{p(\mathbf{a},\mathbf{b}|\mathbf{y})}{p(\mathbf{a}|\mathbf{y})}} = \frac{p(\mathbf{a}|\mathbf{y})}{p(\mathbf{a})} \tag{5.19}$$

$$\propto \frac{p(\mathbf{a},\mathbf{y})}{p(\mathbf{a})} = p(\mathbf{y}|\mathbf{a}). \tag{5.20}$$

Thus, the weight is the marginal likelihood of the model when the component $\mathbf{a}$ is known.

In the general mixture model setting, sampling mixture components from the prior (as in Step 1, above) can potentially lead to estimates that are numerically unstable: When the prior on the mixture components is uninformative (eg, uniform), most of the samples are not highly “important”, and this leads to an inefficient sampling of the posterior (Carlin and Louis, 2009). In Diploffect, however, the prior is the set of diplotype probabilities from HMM, which tends to be relatively well informed for most individuals such that most of the resulting samples are important.

I present two implementations of the IS method: `DF.IS` models the optional family structure term in Eq 5.13 simply as a random intercept representing sibship, ie, as $u_r \sim \mathrm{N}(0, \tau_u^2)$; `DF.IS.Kinship`, an elaboration of `DF.IS`, uses instead a random intercept whose expected covariance is based on the additive relationship (kinship), that is, a polygenic effect after Kennedy et al. (1992): here $\mathbf{u} \sim \mathbf{N}(\mathbf{0}, \tau^2_{kinship}\mathbf{K})$ where $\mathbf{K}$ is the kinship matrix estimated from the pedigree information (Vazquez et al., 2010). Although the kinship matrix provides much richer information about how individuals are related (eg, Cheng et al. (2011)), it also incurs a significantly greater computational cost; increased computational time is also why I do not implement kinship in `DF.MCMC`, whose already significant computation in MCMC sampling becomes impractically slow for repetitive simulation-based assessment when a polygenic term is added.

5.2.8 Partially Bayesian Approximation: `DF.MCMC.pseudo` and `DF.IS.noweight`

In their random effects haplotype model, Durrant and Mott (2010) avoid a fully Bayesian treatment in favor of a partially Bayesian approximation, which estimates the posterior of haplotype effects in Eq 5.9 as

$$p(\boldsymbol{\theta}|\boldsymbol{\Psi},\mathbf{y}) \approx \int p(\boldsymbol{\theta}|\boldsymbol{\Delta},\boldsymbol{\Psi},\mathbf{y})\,p(\boldsymbol{\Delta}|\boldsymbol{\Psi})\mathrm{d}\boldsymbol{\Delta}. \tag{5.21}$$

where, for their model, the effects in $\boldsymbol{\theta}$ include only additive effects $\boldsymbol{\beta}$, additional covariates or structure terms must be absent, and the sampling distribution of the phenotype is restricted to being Gaussian. In the above approximation, the posterior of diplotypes given the phenotype, $p(\boldsymbol{\Delta}|\boldsymbol{\Psi},\mathbf{y})$, is not defined, and the integration is therefore akin to an unweighted multiple imputation. Nonetheless, by limiting the scope and flexibility of the model is this way, Durrant and Mott (2010) are able to derive a fast, direct sampling solution. Moreover, although approximate, their solution would be expected to provide results close to a full Bayesian treatment when $p(\boldsymbol{\Delta}|\boldsymbol{\Psi}) \approx p(\boldsymbol{\Delta}|\boldsymbol{\Psi},\mathbf{y})$, eg, when the QTL's effect is weak or the posteriors of the diplotypes are vague. To explore the utility of this approximation, I implement this approximation applied to both `DF.MCMC` and `DF.IS` methods, respectively as `DF.MCMC.pseudo` and `DF.IS.noweight`. In `DF.MCMC.pseudo`, the sampling of the posterior of $\boldsymbol{\Delta}$ conditional on the current value of $\boldsymbol{\theta}$ in Eq. 5.14 is replaced by a draw from the prior (as in Eq. 5.16); this method was recently used by us in the analysis of immune phenotypes in the Pre-CC (Phillippi et al., 2014). In `DF.IS.noweight`, the IS procedure described for `DF.IS` is modified so that weights are uniform, ie, $w^{(k)} = 1$ for all k; this latter approach is similar to that used in the *Arabidopsis* study of Kover et al. (2009), who instead estimate $\boldsymbol{\beta}$ in an OLS fixed effects regression model.

5.2.9 Non-Bayesian Regression Approximations: `partial.lm`, `ridge.add` and `ridge.dom`

To allow comparison of the mixture model approaches above with approximation that of regress on probabilities, two alternative ROP approaches are considered: a marginal estimator, `partial.lm`; and a ridge regression estimator, implemented in `ridge.add` and `ridge.dom`. The marginal estimator `partial.lm` uses a single predictor linear model to estimate, for each founder haplotype, the effect that haplotype's dosage on the phenotype, ie,

$$\eta_i = \mu_j + \beta_j \left\{ \mathbf{add}(\mathbf{P}_i(m)) \right\}_{ij},$$

where $\boldsymbol{\beta}_j$. This method, which avoids stability any problems related to collinearity in the design matrix by fitting only one effect at time, was used to estimate effects in the Pre-CC study of Aylor et al. (2011b).

A tradition solution for stable simultaneous estimation of all regression parameters under collinearity is ridge regression (Hoerl and Kennard, 1970). In `ridge.add`, ridge regression is applied to the additive ROP model in Eq 5.6, estimating haplotype effects as the value of $\boldsymbol{\beta}$ that minimizes $\sum_i (y_i - \eta_i)^2 + \lambda \boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{\beta}$, where η_i is the linear predictor, and λ is the tuning parameter, which I set by 10-fold cross-validation. In `ridge.dom`, an analogous model is fitted based on the additive plus dominance ROP model of Eq 5.7.

5.2.10 Implementation Details

MCMC-based approaches (`DF.MCMC` and `DF.MCMC.pseudo`) were implemented in R (R Development Core Team, 2011), JAGS (Plummer, 2003), and rjags (Plummer, 2011). JAGS is an open-source general MCMC sampling pack-

Model	Description	ROP
`partial.lm`	Single Haplotype effects Linear Regression.	Yes
`ridge.add`	Ridge Regression with modeling additive effects.	Yes
`ridge.dom`	Ridge Regression with modeling both additive and deviated effects.	Yes
`DF.IS.noweight`	Multiple imputation	No
`DF.IS`	Importance Sampling	No
`DF.IS.kinship`	Importance Sampling including kinship effects	No
`DF.MCMC.pseudo`	Diploffect model with modeling additive and deviated effects, inverse gamma prior and multiple imputation on haplotypes.	No
`DF.MCMC`	Diploffect model with modeling additive and deviated effects.	No

Table 5.2: Summary of the haplotype estimation procedures evaluated in this Chapter.

age; I implemented add-on code to support the partially Bayesian prior sampling of `DF.MCMC.pseudo`. When applying MCMC sampling the Diploffect model, I used a burn-in of 1000 iterations, and 4000 iterations for MCMC sampling, using one sample of every 10 iterations (thinning). Importance Sampling approaches (`DF.IS`, `DF.IS.noweight`, `DF.IS.kinship`) were implemented using the R package INLA (Rue et al., 2009). In each application of the IS methods I used 1000 independent samples directly draw from the haplotype probabilities inferred through HMM model. Ridge regression was performed using the R package GLMNet (Friedman et al., 2010), with tuning parameters selected by 10-fold cross validation. All other analysis was performed in R. For both DF.IS and DF.IS.noweight, I used 1000 independent samples directly draw from the haplotype probabilities inferred through HMM model.

A summary of the procedures evaluated in this study is given in Table 5.2.

5.3 Data and Simulations

I evaluate the ability of Diploffect model to estimate haplotype and diplotype effects by simulation: Using simulated QTL and genotype data, I apply the methods listed in Table 5.2 and compare their ability to both correctly estimate and correctly rank effects. Practical use of the Diploffect model is then illustrated through application to real, previously mapped QTL. Both simulation and application use data from two real populations: the incipient strains of the Collaborative Cross (Pre-CC) (Aylor et al., 2011b), the and the Northport Heterogeneous Stock mice (HS) (Valdar et al., 2006). These data sets are described below.

5.3.1 Pre-CC Dataset

From Pre-CC lines (the description is at Section 4.3.1), diplotype probability matrices are generated by HAPPY (Mott et al., 2000) based on genotype information for 16159 markers across the genome. Also, for real data analysis, I use phenotype data and a mapped QTL location for a binary trait, white spot: This denotes whether or not a white spot is present on the forehead, is observed in both the founder WSB and in 6 out of the 184 Pre-CC mice, and was mapped by Aylor et al. (2011b) to a QTL at 92.0 MB on Chromosome 10.

5.3.2 Heterogeneous Stock (HS) Dataset

The Heterogeneous Stocks is an outbred population of mice also derived from 8 inbred strains: A/J (AJ), AKR/J (AKR), BALBc/J (BALB), CBA/J (CBA), C3H/HeJ (C3H), C57BL/6J (B6), DBA/2J (DBA) and LP/J (LP). I used data from the study of Valdar et al. (2006), which uses mice from approximately generation 50 of the cross, and comprises genotypes and phenotypes for 1762 mice from 180 families, with family sizes varying from 8 to 48. From this population I use diplotype probability matrices generated by also HAPPY based on genotype information for 10148 markers across the genome. For application to real data, I use two phenotypes: total total cholesterol (CHOL: 1656 observations), mapped by Valdar et al. (2006) to a QTL at 171.5-172.0Mb on chromosome 1; and the total startle time to a loud noise (Fear Potentiated Startle; FPS; 1508 observations), which in the same study was mapped to a QTL at 91.37-92.62Mb on chromosome 15.

5.3.3 Informativeness of Haplotype Reconstruction in Pre-CC and HS

Diplotype probabilities in the Pre-CC dataset are more informative than those in the HS. This is because the Pre-CC was not only genotyped with more markers than the HS but also contains fewer recombinants per chromosome, thus benefitting from a greater genotyping density relative to expected recombination fraction. This difference in informativeness is depicted in Figure 5.4, which shows for each dataset the distribution of per-locus scaled Selective Information Content (SIC; as used in, eg, Ronnegard and Valdar (2011b)). SIC, equivalent to a rescaling of Shannon's entropy, ranges from 0, denoting all individuals are uninformative at a locus, to 1, denoting all individuals have diplotype assigned with certainty. Locus information varies in both populations, and the Pre-CC contains a degree of uncertainty even in among those loci that are most informed; nonetheless, the HS is seen to be by far the more challenging target for accurate inference of QTL effects.

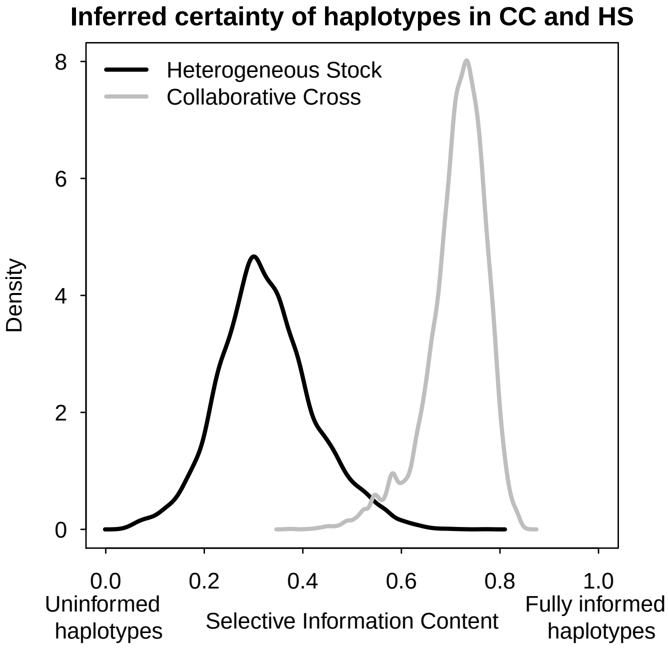


Figure 5.4: Certainty of inferred diplotype assignments across all marker loci in the Pre-CC and HS.

5.3.4 Simulating QTL effects

The ability of the Diploffect-base methods to estimate and rank haplotype and diplotype effects is assessed by simulation: applying those methods, and their competitors listed in Table 5.2, to simulate single QTL for which the true effects are known. This is performed first using Pre-CC data, in which estimation of haplotype (ie, additive) effects is emphasized, potentially in the presence of dominance from residual heterozygotes; then, separately, using the HS data, which emphasizes estimation of diplotype effects that arise from both additive and dominance. In either population, simulation of QTL involves four basic steps: selecting a locus; assigning true diplotypes; assigning effects to each diplotype and thereby generating "pure" (ie, expected) phenotypes for each individual; and, adding individual phenotypic noise, which may include genetic background. These steps are detailed below.

In a given simulation trial, for both Pre-CC and HS simulations, a single locus is selected to be the QTL at random from 50 markers evenly distributed across the entire genome. At that QTL, a "true" diplotype at the QTL is then assigned to each individual based on a random draw from their diplotype probability matrix (Eq 5.5; as in, eg, Durrant and Mott (2010)). The diplotype state for each individual is then used to calculate for each individual i a QTL effect q_i, by combining additive effects $\boldsymbol{\beta}$ and dominance effects $\boldsymbol{\gamma}$ using the linear predictor in Eq 5.2. Those additive and dominance effects are generated as follows. Additive effects $\boldsymbol{\beta}$ in one simulation trial are generated in one of two ways: as a binary vector, in which a SNP-like effect distinguish two groups of founder haplotypes, and $\boldsymbol{\beta}$ is drawn at random from a representative set of 25 such vectors (eg, (0,0,0,0,1,1,1,1), (0,1,1,1,1,1,1,1)); or, as a random draw from a multivariate normal distribution, $\boldsymbol{\beta} \sim \mathrm{N}(0, \mathbf{I})$. In total, 50 types of additive effects are used in the simulation study (see Supplemental Methods). If dominance deviation effects are required, they are drawn as $\boldsymbol{\gamma} \sim \mathrm{N}(0, \mathbf{I})$, such that variance contributed additive and dominance effects are equal. Each individual's QTL effect q_i is then combined with an individual random noise term $\varepsilon_i \sim \mathrm{N}(0, 1)$ and genetic background term u_i (defined below) to give a simulated phenotype $y_i = aq_i + bu_i + c\varepsilon_i$, where a, b and c are constants used to adjust relative contributions of QTL, background and noise to the phenotype. QTL effect sizes are set relatively large, accounting for 10%, 20%, 30% and 40% of the phenotype variance. In the Pre-CC simulations, the remaining variance is entirely due to individual noise — in this population, with one individual per line and all lines drawn independently, individuals are evenly related (in expectation), making genetic background effects negligible. In the HS population, where individuals are related by differing degrees, 50% of the phenotypic variance was contributed by genetic background and the remainder by individual noise; in this case, genetic background was simulated by drawing u_i for $i = 1, \ldots, n$ jointly as $\mathbf{u} \sim \mathrm{N}(\mathbf{0}, \mathbf{K})$, where $\mathbf{K}$ is the kinship matrix calculated using EMMA (Kang et al., 2008).

In summary, a simulation trial chooses one marker as the QTL from 50 different candidate markers, one type of additive effects vector as the "true" effects from 50 different types of additive effects, and one QTL size from 4 different sizes. Therefore, for each population, I conduct 10000 ($= 50 \times 50 \times 4$) simulation trials with only additive effects, and another 10000 simulation trials with both additive and dominant effects.

5.3.5 Evaluating Performance of Effect Estimation

Methods are evaluated by two criteria: how far estimates are to the truth (prediction error), and how accurately they capture rank ordering (rank accuracy). Prediction error was judged by an adjusted version of the mean-squared error (MSE). Specifically, let $\boldsymbol{\theta}$ denote the K-vector of simulated effects (the target) after mean-centering, and let $\hat{\boldsymbol{\theta}}$ be the K-vector of point estimates for the target, also after mean centering. The definition of $\hat{\boldsymbol{\theta}}$ depends on the estimation method used: for Bayesian or partially Bayesian methods in Table 5.2 (`DF.IS`, `DF.IS.kinship`, `DF.IS.noweight`, `DF.MCMC`, and `DF.MCMC.pseudo`) it is defined as the posterior mean; for the remaining methods (`partial.lm`, `ridge.add`, and `ridge.dom`) it is the standard point estimate (ie, that maximizing the likelihood or penalized likelihood). Prediction error is then defined as the difference between target and estimate, normalized by the variance of the target:

$$\text{prediction error} = \frac{(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})^{\mathsf{T}}(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})}{K \times \text{Var}(\boldsymbol{\theta})\,.} \tag{5.22}$$

The set of effects included in the target $\boldsymbol{\theta}$ differs according to the simulation setup. For the Pre-CC, which is almost inbred, the target includes only the haplotype (additive) effects, ie, $\boldsymbol{\theta} = \boldsymbol{\beta}$; dominance effects may be present, but their occurrence in the data — while enough to derail inference of $\boldsymbol{\beta}$ — is too sporadic for meaningful estimation and are therefore considered as a type of noise in the Pre-CC. For the HS, many heterozygous diplotype states will be present at a given QTL, although overall some diplotype states may be absent; for this population, the target therefore includes the $J \times (J+1)/2$

vector of diplotype effects, ie, $\boldsymbol{\theta} = \boldsymbol{\delta}$.

In addition to prediction error, which is primarily motivated by how well effects estimated for the QTL would predict phenotype in new individuals, I also examine the rank, which is most relevant to how effects are interpreted (eg, when identifying high and low strain effects); rank accuracy is calculated using Spearman's rank correlation of the target $\boldsymbol{\theta}$ with the estimate $\hat{\boldsymbol{\theta}}$.

5.4 Results

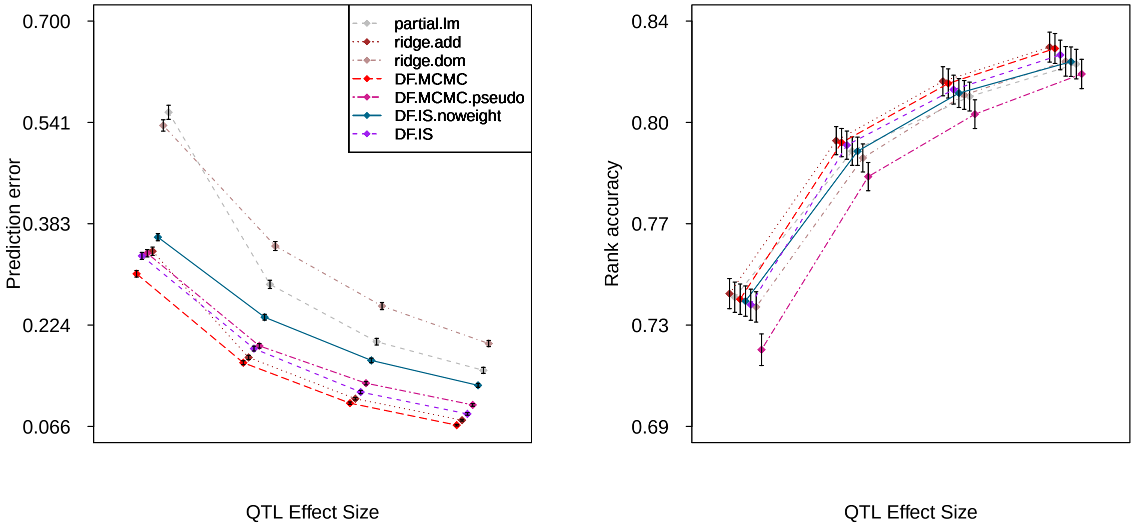
Methods to estimate QTL effects, including those based on Diploffect model, were evaluated by simulation. Simulations were performed on two datasets (Pre-CC and HS) under two settings (additive effects only, and additive plus dominance effects), giving rise to four distinct simulation studies. In each study, to assess how methods improved with greater signal to noise, the magnitude of the QTL effect was varied from 10% to 40% of the phenotypic variance. Estimation of those effects was judged by two criteria: prediction accuracy, which is most relevant to estimating correctly the magnitude of effects, and therefore especially relevant to out of sample predictions; and rank accuracy, which is most relevant to interpretation that focuses on how the effects of different founder haplotypes (or founder-pair diplotypes) are ordered. Interval estimates of QTL effects were not considered in the simulations because for the non-Bayesian methods examined they are undefined; however, examples of interval estimates are provided later in three example applications estimating effects for previously identified QTL.

5.4.1 Pre-CC Simulations: Estimation of Additive Effects

All seven methods listed in Table 5.2 were evaluated for their ability to estimate simultaneously 8 haplotype effects at simulated QTL in the Pre-CC. Their accuracy in estimating the relative numerical distance between effects (prediction accuracy) is given in Figure 5.5(a). This shows strongest performance by: fully Bayesian implementations of Diploffect (`DF.MCMC` best of all; then `DF.IS`) its partially Bayesian MCMC implementation (`DF.MCMC.pseudo`); and then additive-effect only ridge regression (`ridge.add`), which 10% worse than the fully Bayesian methods but at larger QTL effect sizes is better than the partially Bayesian methods. Worst performance by far is seen for the single predictor linear model `partial.lm` and the 36-parameter ridge method `ridge.dom`; both of these seem to produce erratically dispersed estimates, with this exacerbated by smaller effect QTL. Although the different methods vary considerably in their ability to estimate magnitude of effects, they are hardly distinguished in their ability to determine relative rank: In Figure 5.5(b), the all methods are insignificantly different, with the exception of the partially Bayesian `DF.MCMC.pseudo`, which significantly (although not substantially) underperforms the others. In general: fully Bayesian methods either equal or outperform both partially Bayesian methods and the non-Bayesian methods; highly parameterized ridge and the single predictor regression inflate estimates but preserve order; and 8-parameter ridge performs well in both prediction and rank accuracy.

5.4.2 Pre-CC Simulations: Estimation of Additive Effects in the Presence of Dominance

The simulations above were repeated, but this time with QTL simulated to include not only additive effects but also effects of dominance. Dominance effects have few opportunities to manifest at typical locus in the Pre-CC because that population is mostly inbred. Nonetheless, residual heterozygosity is present at many loci, and so dominance can, if unmodeled, potentially disrupt estimation of additive effects. To examine this phenomenon, methods were evaluated based on their estimated additive (haplotype)



(a) Prediction error

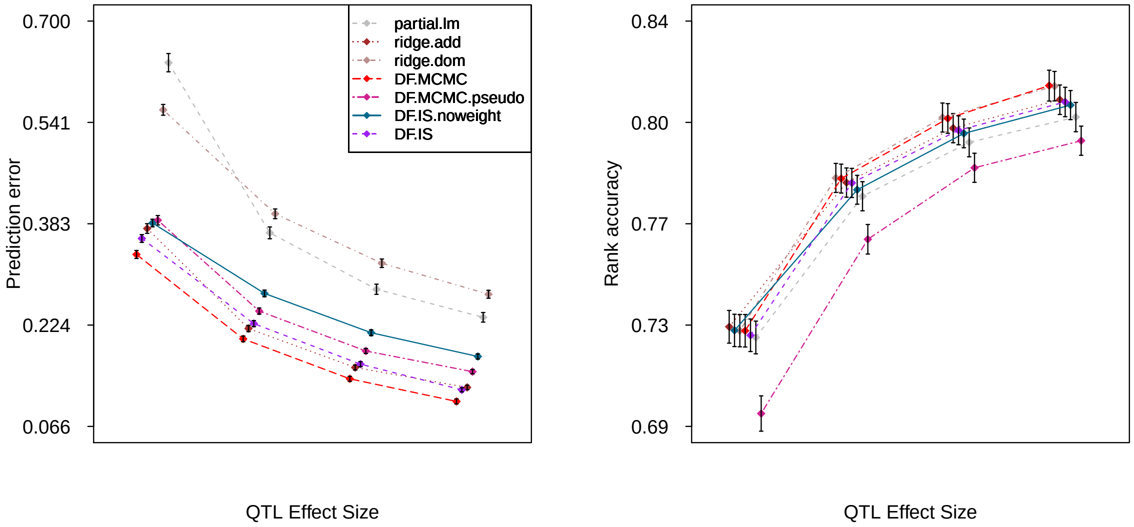
(b) Rank accuracy

Figure 5.5: Estimation of additive effects for a simulated additive-acting QTL in the Pre-CC population, judged by a) prediction error, and b) rank accuracy. For a given combination of QTL effect size and estimation method, each point indicates the mean of the evaluation metric based on 2500 simulation trials, and each vertical line indicates the 95% confidence interval of that mean. Points and lines are grouped by the corresponding QTL effect sizes and also are shifted slightly to avoid overlap. At the same QTL effect size, the left to right ordering of the methods reflects relative performance of better to worse.

effects, albeit in the presence of disruptive dominance. As shown in Figure 5.6, the addition of dominance, although increasing prediction error and decreasing rank accuracy, did not visibly change the relative performance of the methods. Interestingly, the prediction error of `ridge.dom` remains on the order of 80% worse than `ridge.add` despite its inclusion of (potentially useful, in this case) dominance parameters.

5.4.3 Updating of Diplotype Probabilities in the Pre-CC

In `DF.MCMC`, diplotype state for each individual is a modeled latent parameter, and, like other modeled latent parameters, has both a prior and a posterior distribution: The prior in this case is the diplotype probability from haplotype reconstruction, which is



(a) Prediction error

(b) Rank accuracy

Figure 5.6: Estimation of additive effects for a QTL simulated to have both additive and dominant effects in the Pre-CC population. Symbols are defined as in Figure 5.5.

ignorant of phenotype; the posterior is estimated from the MCMC samples, is cognizant of phenotype, and update the original probability in light of that phenotypic information. Provided adequate MCMC mixing occurs and enough MCMC samples are taken, this updating process should result in an increased posterior probability placed on the true underlying diplotype state of that QTL in that individual. Since in the simulations I generate these true diplotype states, the extent to which diplotype probabilities are improved can be observed. This is illustrated for one individual's prior and posterior diplotype probabilities in Figure 5.2 of Section 5.2.6. The improvement across all individuals was quantified using a summary statistic, the True Diplotype Improvement (TDI), which measured the average difference between posterior and prior probability for each underlying diplotype $jk[i]$ for each individual i, defined as

$$\text{TDI} = \frac{1}{n}\sum_{i=1}^{n}\left[p(D_i(m)_{jk[i]}|\mathbf{y}) - P_i(m)_{jk[i]}\right] . \tag{5.23}$$

In the Pre-CC simulations, averaged across trials involving additive only effects the average TDI was 0.008 (95% CI: 0.007-0.009); averaged across all trials involving

Population	Pre-CC	Heterogeneous Stock
partial.lm	0.056	2.72
ridge.add	0.124	2.80
ridge.dom	0.151	2.92
DF.MCMC.pseudo	27.96	NA
DF.MCMC	92.77	NA
DF.IS.noweight	580.7	3727
DF.IS	580.9	3724
DF.IS.kinship	NA	16231

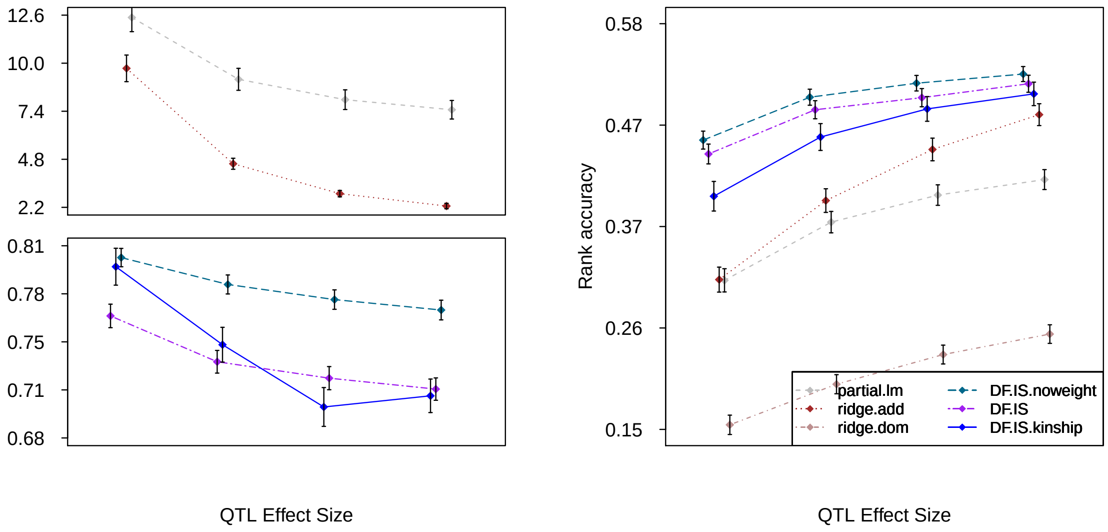
Table 5.3: A table for running time (seconds) of different models.

additive plus dominance effects, the average TDI was 0.015 (95% CI: 0.014-0.015).

5.4.4 HS Simulations: Estimation of Additive Effects

Six of the seven methods in Table 5.2, all except DF.MCMC and DF.MCMC.pseudo, were evaluated on their ability to estimate simultaneously 36 diplotype effects for additive-effect QTL simulated in the HS population. The MCMC methods were excluded from this comparison because they were too slow: The time required for acceptable MCMC convergence on this relatively large dataset (1762 individuals) made performing 2500 independent analyses under each of four conditions unfeasible (see 5.3). Diploffect models were therefore represented by the importance samplers: DF.IS, DF.IS.noweight, and DF.IS.kinship. Of these, genetic background effects arising from unequal relatedness are represented in two ways: DF.IS.kinship uses a full polygenic model based on the genotype-inferred kinship matrix; DF.IS and DF.IS.noweight approximate this polygenic model with a random intercept for sibship — an approximation that reduces their running time by more than four-fold (Table 5.3).

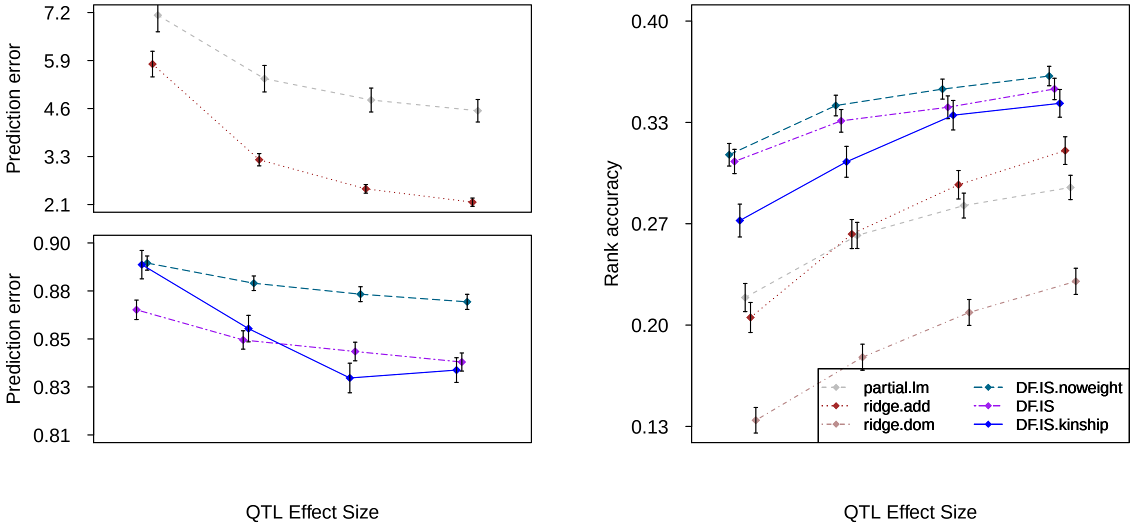
As shown in Figure 5.7, under this setting where sample size is greater but diplotype state is less certain, methods based on the Diploffect model strongly outperform the regression-on-probability based competitors. This somewhat expected: In the face of uncertainty, ROP will lead to wildly inflated estimates and poor prediction accuracy (see discussion of Table 5.1 in Methods). Ridge regression in this context should improve considerably over standard least squares multivariable regression. Nonetheless, `ridge.add` produced effect estimates that were considerably dispersed relative to their true values, and `ridge.dom` performed so poorly (prediction error > 500) that it had to be excluded from Figure 5.8(a) for the purposes of legibility. Among the Diploffect models, the fully Bayesian methods `DF.IS` and `DF.IS.kinship` significantly outperform the partially Bayesian `DF.IS.noweight` on prediction accuracy, but this advantage is reversed for rank accuracy, where `DF.IS` is slightly worse, and `DF.IS.kinship` is significantly worse (discussed more below).



(a) Prediction error

(b) Rank accuracy

Figure 5.7: Estimation of diplotype effects for an additive-only QTL simulated in the HS. Symbols are defined as in Figure 5.5



(a) Prediction error

(b) Rank accuracy

Figure 5.8: Estimation of diplotype effects for QTL simulated to have both additive and dominance effects in the HS. Symbols are defined as in Figure 5.5

5.4.5 HS simulations: Estimation of Additive and Dominance Effects

The simulations described for the HS above were repeated, but this time with QTL simulated to include effects of both additive and dominance. Relative performance of the five methods remained about the same; absolute performance on prediction accuracy worsened a little for the Diploffect-based methods, and absolute performance on rank accuracy dropped sharply by 0.1 (correlation scale 0 to 1) for all methods. As in the simulations with additive-only QTL, modeling genetic background using a kinship-specified polygenic effect, as in `DF.IS.kinship`, is not clearly superior to using a sibship-based approximation; indeed, at least in this context, it performs significantly worse on rank accuracy while requiring substantially more computation. I speculate that this relative robustness of the sibship approximation could reflect either the breeding structure of the HS, which, perhaps because of its circular mating structure, leads kinship to be well-approximated by sibship, or/and computational efficiencies associated with estimating polygenic effects (see also below) rather any advantage of sibship approximations in general.

5.4.6 Efficiency of Importance Sampling in the Pre-CC and HS

Unlike the MCMC-based methods, the methods based on importance sampling are non-iterative: Although in both cases inference benefits from more posterior samples, for the IS methods it does not also require the potentially slow and unpredictable convergence of an MCMC chain. Nonetheless, when sampling such a large parameter space (simultaneously $\boldsymbol{\theta}$ and $\boldsymbol{\Delta}$ in Eq 5.8), an IS procedure that reweights samples from the prior (as IS scheme does, albeit partially) can be highly inefficient when that prior is uninformed; in particular, a large number of samples drawn from the prior may, after reweighting, translate into a comparatively tiny number of samples from the posterior. I investigated the extent of this sampling inefficiency using the effective sample size (ESS) metric of Liu et al. (2001):

$$\text{ESS} = \frac{(\sum_k w^{(k)})^2}{\sum_k (w^{(k)})^2}$$

where $w^{(k)}$ is the weight for the kth sample (see Models and Methods). As shown in Figure 5.9, `DF.IS` applied to HS leads to a much smaller ESS than the `DF.IS` applied to the Pre-CC, reflecting both the greater size of the posterior space for the HS and the greater uncertainty present in the HS diplotype probabilities. Although the ESS metric can be misleadingly high when, eg, all draws are far from the posterior, when it is low it implies that estimation will be inefficient (and potentially high variance) because the estimated posterior well-informed by only a few samples. This could explain why even though `DF.IS` is better than `DF.IS.noweight` in prediction error, `DF.IS` is outperformed by `DF.IS.noweight` in rank correlation, suggesting that `DF.IS` may be inefficient under a weakly informed diplotype prior. Conversely, under such weakly informed diplotype priors, `DF.IS.noweight` can potentially lead to more stable (albeit less informed) inference because all draws are weighted equally.

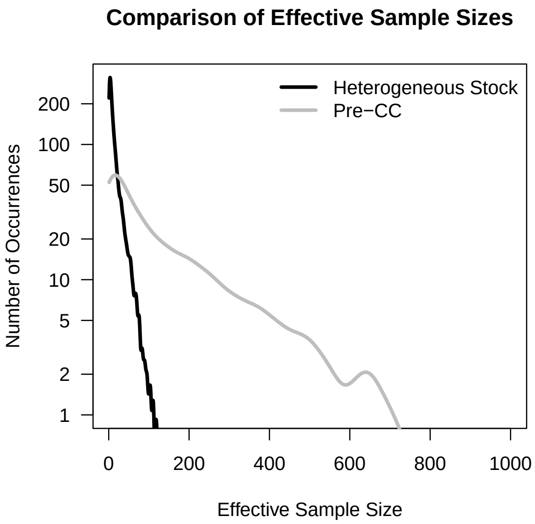


Figure 5.9: Density plot of the Effective Sample Size of posterior samples for the `DF.IS` method (maximum possible is 1000) applied to HS and Pre-CC. ESS measures how efficiently the true posterior is sampled by `DF.IS`. Distribution is based on IS samples from 5000 independent simulations.

5.4.7 Haplotype Effects on A Binary Outcome: White Head-spotting in the Pre-CC

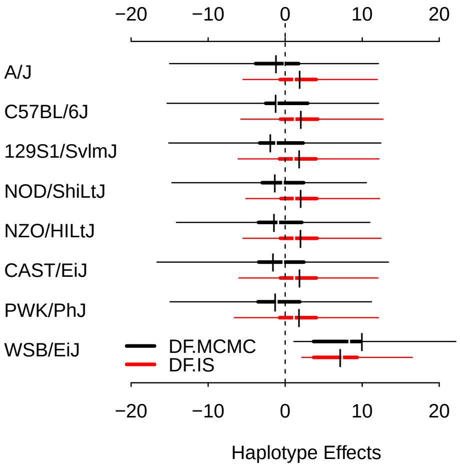
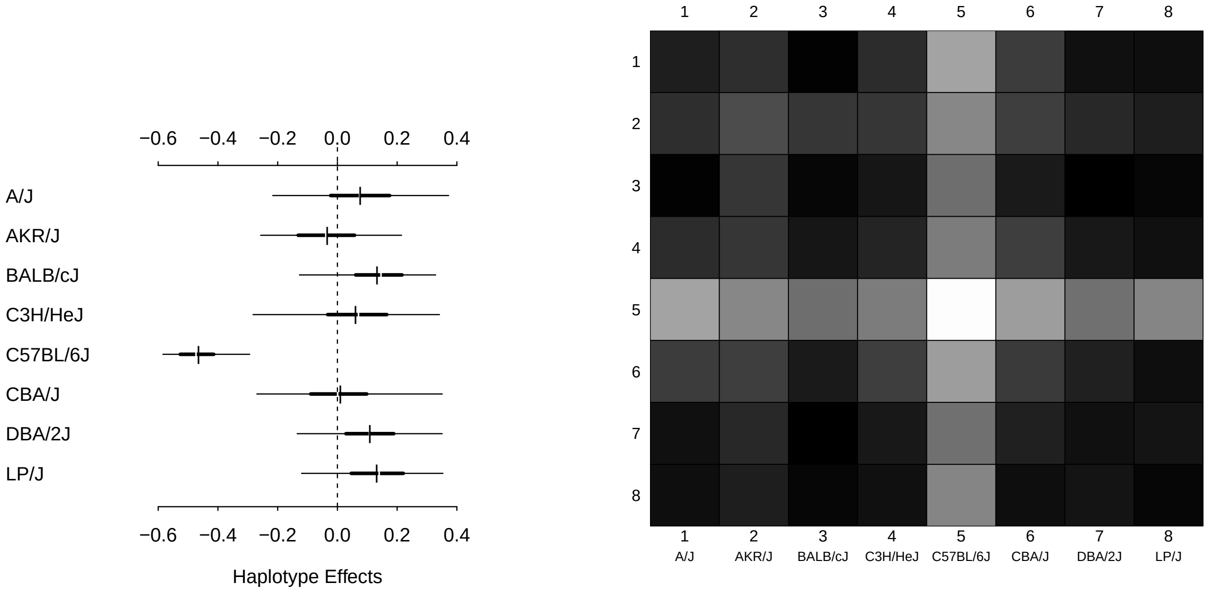


Figure 5.10: Highest posterior density intervals for the haplotype effects of the binary trait white-spotting in the Pre-CC

The Pre-CC study of Aylor et al. (2011b) identified a Mendelian trait locus on chromosome 10 (at 92.0 Mb) for white head-spotting. White head-spotting is a characteristic of the inbred CC founder strain WSB, and this phenotype was visibly present in 6 out of the 184 Pre-CC mice. Because the identified locus was dominant Mendelian, associated with the presence of either one or two WSB haplotypes, it was straightforward to identify by LD mapping using a haplotype dosage ROP model as in Eq 5.6. Estimating meaningful strain effects was not, in this case, necessary, because the effect was obvious. It would, however, have been awkward statistically, because proper treatment of the binary outcome is most naturally modeled as a binary logistic regression, which in a standard maximum likelihood estimation would have quickly become problematic due to separation (see, eg, Gelman and Hill (2007)). Because Diploffect is both defined for as generalized linear and includes automatic variable shrinkage, strain effects for white spot can be modeled without further development. In Figure 5.10, I plot 95% highest posterior density (HPD) intervals for all haplotype effects at the QTL estimated by both `DF.MCMC` and `DF.IS`. Here, HPDs for `DF.IS` are calculated by Rao-Blackwellization: Marginal densities for each haplotype effect are estimated at each importance sample, and these are subsequently reweighted to give a mixture density from which the HPD interval is derived. Both models report a similar result for this QTL: The non-WSB posteriors are similar to each other and broad, reflecting high uncertainty about the relative effects of these strains; the WSB posterior distributions is shifted above the others, reflecting its positive effect. The HPD of the contrast WSB vs the other strains, calculated by applying this contrast to each MCMC sample from `DF.MCMC`, is 1.35-36.18, further reflecting the positive effect of the WSB haplotype but also the fact that uncertainty about this effect remains because the sample size is not infinite.



(a) The HPD intervals of the effects at the QTL of FPS.

(b) The predicted diplotype effects of FPS at QTL.

Figure 5.11: Haplotype and diplotype effects estimated by `DF.IS` for phenotype FPS in the HS

5.4.8 Haplotype and Diplotype Effects at QTL in the HS: Fear Potentiated Startle (FPS) and Total Cholesterol (CHOL)

To demonstrate Diploffect-based estimation of additive and dominance effects, I examined two previously mapped QTL from the HS mapping study of Valdar et al. (2006). The first QTL is for Fear Potentiated Startle, a conditioned test of anxiety (see Solberg et al. (2006) and refs therein), located between 91.37-92.62 Mb on chromosome 15. The `DF.IS` procedure was applied to the central marker interval of this QTL (rs3722990 - rs3716673). Marginal posteriors for all effects were calculated as above. For legibility, I show HPD intervals for haplotype effects only, in Figure 5.11(a). Dominance effects, which comprise 28 deviations from the additive haplotype model, are harder to graph intuitively; instead I plot the posterior predictive means of the 36 possible diplotype effects, ie, $E(\boldsymbol{\delta}|\mathbf{y})$, as a symmetric grayscale matrix in Figure 5.11(b).

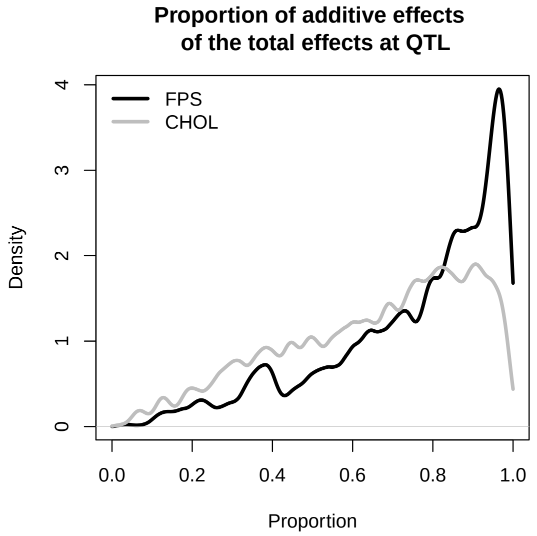
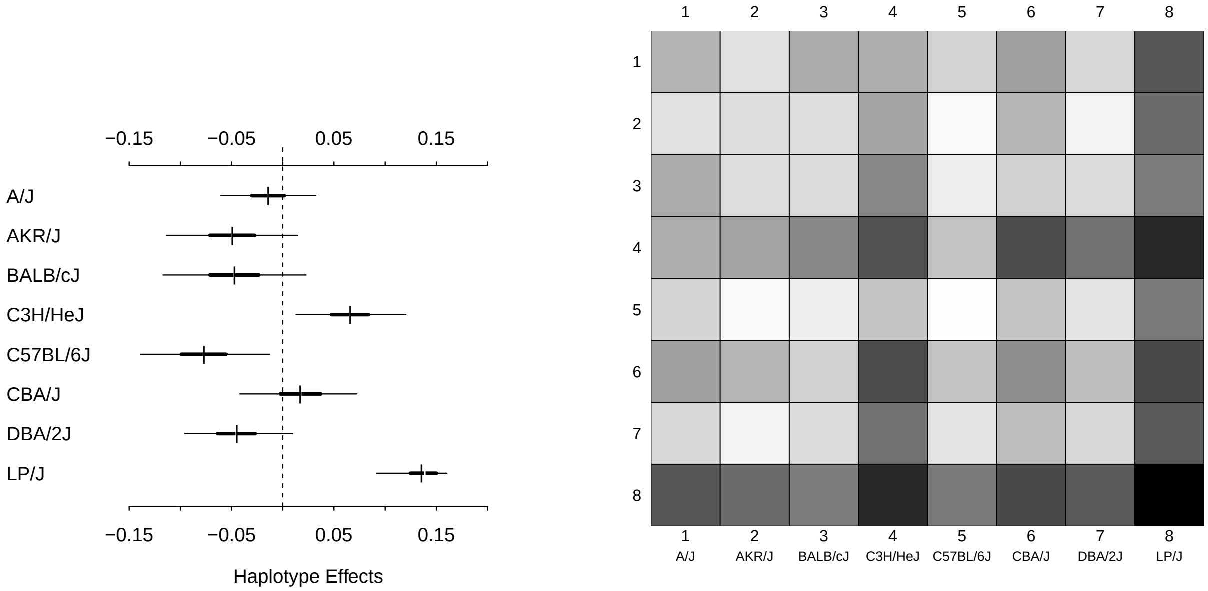


Figure 5.12: Posteriors of the fraction of effect variance due to additive rather than dominance effects at QTL for phenotypes FPS and CHOL in the HS dataset

Both plots suggest that effects are driven by C57BL/6J, and the consistent banding pattern of the diplotype effect plot suggests these effects are mainly additive. The degree of additive vs dominance effects is quantified further in Figure 5.12, which gives the posterior distribution of the fraction of effect variance due to additivity, that is, ie, $p(\pi_{\text{add}}|\mathbf{y})$ where $\pi_{\text{add}} = \tau^2_{\text{add}}/(\tau^2_{\text{add}} + \tau^2_{\text{dom}})$. As expected for a ratio defined using hyperparameters, this posterior is relatively broad; but it nonetheless has a clear maximum near 1, with posterior mean of 90.3%, suggesting that additive effects predominate.

The second QTL examined in the HS was for total cholesterol concentration (CHOL), located between 171.15-171.51 Mb on chromosome 1. As above, the `DF.IS` procedure was applied to the central marker interval of the CHOL QTL (rs13476229-rs3657320) to give HPD intervals for haplotype effects (Figure 5.13(a)) and point estimates of diplotype effects (Figure 5.13(b)). Unlike the FPS QTL, the HPD intervals for CHOL cluster into three different groups: the highest effect from LP, a second group comprising C3H and CBA with positive mean effects, and the remaining five strains having negative effects. This pattern is consistent with a multiallelic QTL, potentially



(a) The HPD intervals of the effects at the QTL of CHOL.

(b) The predicted diplotype effects of CHOL at QTL.

Figure 5.13: Haplotype and diplotype effects estimated by `DF.IS` for phenotype CHOL in the HS

arising through multiple, locally epistatic biallelic variants. In the diplotype effect plot (Figure 5.13(b)), although most of the effects are additive, off-diagonal patches provide some evidence of dominance effects: In particular, the haplotype combinations AKR × DBA and C3H × CBA deviate from the banding otherwise expecting under additive genetics. The fraction of additive effect variance for CHOL has a posterior mean of 82.0% and, as shown in Figure 5.12, a posterior distribution far less concentrated around high additive effects.

5.5 Conclusion

I present here a statistical model and associated computational techniques for estimating the effects of alternating haplotype composition at QTL detected in multiparent populations. The statistical model of Diploffect is intuitive in its construction, connecting phenotype to underlying diplotype state through a standard hierarchical regression model. Its chief novelty, and the source of greatest statistical challenge, is that diplotype state, although efficiently encapsulating multiple facets of local multilocus variation, cannot be observed directly, and is typically available only probabilistically — meaning that statistically coherent and predictively useful description of QTL action requires estimating effects of alternating haplotype composition from data where composition is itself uncertain.

Nonetheless, use of the Bayesian procedures proposed here has several potential drawbacks. Foremost is computation time: Although the modified slice sampler (`DF.MCMC`; Section 5.2.6) makes MCMC sampling of both diplotypes and effects feasible, it is nonetheless highly computationally intensive. For large outbred populations, especially those with a high degree of diplotype uncertainty (which can be examined using SIC as in Figure 5.4, I recommend `DF.IS` over `DF.MCMC`. For either method, a high degree of diplotype uncertainty and weak QTL effects results in computational inefficiencies because the posterior distribution that must be traversed (in MCMC) or sampled (in IS) is much more diffuse: for `DF.MCMC` this means convergence must be carefully monitored; for `DF.IS`, this means many more samples must be taken to achieve a reasonable picture of the posterior.

CHAPTER 6

CONCLUSION

This chapter concludes the dissertation by summarizing my contributions and proposing some future work.

6.1 Summary of Contributions

Multiparent crosses have demonstrated their uniqueness in genetics: They are as diverse as human populations in terms of encoding genetic factors in their genomes, while they are still maintained in a well controlled manner, ensuring replicability of biological experiments. Existing general purpose computational genetic tools may not be able to gauge sufficient information from multiparent crosses because they are not specifically designed for such populations, neglecting several important features in multiparent crosses (e.g. finite number of founders, precisely assembled genomes of the founders, special pedigree design); some key studies such as haplotype effect estimation also lack corresponding computational methods, limiting the scope of how multiparent crosses may be applied.

I have developed a series of methods for solving several computational and statistical problems in analyzing data from multiparent crosses, including RNA-Seq assembly and quantification, QTL mapping and haplotype effects estimation. These methods are motivated by the collaborating projects when I conducted for analyzing genomic data for Collaborative Cross. Therefore, the special characters of multiparent crosses are utilized by these methods, and this dissertation proves they can be used to empower and enrich the analysis. While some method such as Diploffect is specifically designed and only applicable for multiparent crosses, the others including GeneScissors, RNA-Skim and HTreeQA can also be applied to other types of populations including human population. My particular contributions are listed as follows.

- **Connecting existing genome information to computational tools.**

  Many approaches, including Cufflinks (Trapnell et al., 2010), RSEM (Li and Dewey, 2011), etc., start to use existing information such as annotated transcriptome to help improve the efficiency of the methods. However, only the sequences of transcripts are used by these methods, meta information such as the types of genes is not considered. In addition, few methods refine the existing information first for developing rapid computational methods. It is sensible that these methods can be further improved by incorporating more existing knowledge into themselves, especially when data analysts apply these methods on well-studied populations that are complemented with enriched information, including multiparent crosses; it is nonetheless unclear how to achieve this goal.

  In this thesis, I've demonstrated two new ways to integrate the existing knowledge: GeneScissors (Chapter 2) utilizes the annotated meta information from transcriptome to validate the result, while RNA-Skim (Chapter 3) analyzes the whole transcriptome to find sig-mers to speed up the RNA-Seq quantification. Compared with existing methods, GeneScissors is able to generate more trustworthy results, and RNA-Skim is much faster by using a smaller set of sig-mers instead of using the whole transcriptome.

- **Using novel concepts for genomic analysis**

  In this dissertation, I've proposed several unique concepts for designing and implementing computational genetics methods on RNA-Seq data or QTL mapping, including the sharing graph and fragment attractors (Chapter 2), which are from an idea that leverages results from both aligner and assembler to determine the context of a cluster of alignments at the same genome location, the sig-mers (Chapter 3), which is a type of k-mers that uniquely exist in a subset of transcripts, and the tri-state semi-perfect phylogeny trees (Chapter 4) which are phylogeny trees built from heterozygous genotypes. GeneScissors, RNA-Skim and HTreeQA are designed by the essences of these original and unconventional concepts, and they have successful applications on multiparent crosses and exhibit promising results.

- **Correcting errors in RNA-Seq assembly based on the context**

  Most of existing RNA-Seq analysis pipelines separate aligning and assembling as two independent steps, and the common approach in the aligner attempts to remove suspicious alignments before the assembly. However, evidence has shown that this choice causes failures of assembling genes and reports of unexpressed genes in the transcriptome because the corresponding alignments have been removed by the premature error correction in aligners (Chapter 2).

  GeneScissors recognizes this situation and detect errors produced by it. In the opposite of removing alignments by just basing on alignments of individual fragments, which is a strategy commonly used in RNA-Seq aligners, however, GeneScissors never removes a single alignment in the alignment step, and use fragment attractor to collect “context” information about the alignments. If the alignments of a set of fragments are from a region with full of mismatches, and these fragments can also be aligned to another region without mismatches — which is a sufficient evidence that the former alignments are erroneous, the former ones are removed by GeneScissors. And GeneScissors also uses other information such as number of exons and fractions of shared fragments of a specific transcripts to assist identify errors in RNA-Seq pipelines. As there are a couple of features considered by GeneScissors, it constructs the sharing graph of transcripts reported by existing RNA-Seq pipelines, and builds a Support Vector Machine based graph classification method to identify false nodes in the graph. In this way, the error correction step is postponed until the assembly step finishes, using information of clusters of alignments instead of each individual alignment to identify false alignments and yielding more accurate results than the error correction method used in the existing pipelines.

- **Improving the RNA-Seq quantification performance by 10-fold**

  RNA-Seq quantification typically required hours if not days to analyze a single individual, until the recent development in alignment-free methods that uses k-mers instead of alignments of fragments to quantify the transcripts, delivering the result at the same accuracy yet in a much shorter time. Meanwhile, some fact indicates that the RNA-Seq data is highly redundant, and lots of computations in RNA-Seq tools are wasted to process the redundant information. RNA-Skim is a method that can avoid unnecessary computation by using a much smaller yet still informative set of k-mers — sig-mers — yielding another 10-fold speed-up comparing with Sailfish. In order to achieve this goal, RNA-Skim provides a complete solution that can determine the best clusters of transcripts ensuring that most of transcripts are covered by sig-mers, efficiently finding and selecting sig-mers using bloom filters, rapidly counting all sig-mers in the RNA-Seq data using the rolling hash method, and fast and accurately quantify all transcripts using a statistical model fitted by an Expectation-Maximization algorithm. RNA-Skim significantly saves computational resources used for RNA-Seq quantification, allowing biologists to get the transcript abundances in a much shorter time and saving them times to focus on the follow up analysis such as RNA-Seq differential analysis.

- **Discovering QTLs using tri-state semi-phylogeny trees for heterozygous populations.**

A perfect phylogeny tree reflects the ancestor information, is an ideal representation for the relatedness of individuals at the haplotype level, and it has been widely used in the QTL mapping methods. Since the perfect phylogeny tree can only be constructed on the haplotypes, for diploid populations, the individual with heterozygous sites becomes two different leaves on the perfect phylogeny trees, violating the independent assumption of the statistical tests such as ANOVA which are commonly used in the phylogeny based QTL mapping. I proposed a tri-state semi-perfect phylogeny trees (Chapter 4) to allow that each individual is presented as one leaf node in the constructed phylogeny tree, permitting the validity of the assumption used in the underlying statistical test. From this, I developed a novel QTL mapping method named HTreeQA, extending the previous work of TreeQA, to provide the capability to detect a wide set of effects including additive, dominant and over-dominant QTLs.

- **Estimating Haplotype effects with statistically valid interval estimates**

  As emphasized in this dissertation, one advantage of multiparent crosses is that the number of founders is limited and we already have relatively complete information about the founders, therefore, the QTL also has a limited possible genetic states, enabling investigators to enumerate each of them to understand how they contribute the phenotypes. However, few of existing methods take advantage of this unique feature, restraining users from understanding how founders contributes effects to the phenotypes at the QTLs.

  Diploffect (Chapter 5) is one of the initial attempts to combine the structure of multiparent crosses, the inferred haplotypes and the discovered QTLs to reveal the underlying complicated effects of the founders to the phenotypes by joint modeling the haplotype distributions and the effects of haplotypes at the QTL. I frame this problem as a Bayesian integration, in which both diplotype state and QTL effects are latent variables to be estimated, and provide two computational approaches to solving it: one highly flexible but also heavily computationally demanding, based on MCMC; the other less flexible but where the computation required, although still somewhat expensive, is more predictable. Importantly, in theory and simulation, I describe how simpler, approximate methods for estimating haplotype effects relate to Diploffect, and how the trade-offs they make can affect inference.

The prime comparison, representing the most accessible competitor of Diploffect, is with approaches that directly regression on the diplotype probabilities themselves (or functions of them, such as the haplotype dosage) rather than modeling the latent states those probabilities represent. In the context of QTL detection, where the need to scan potentially large numbers of loci makes fast computation essential, such ROP-based approaches are, in my view, typically well-justified and often the only practical solution. For estimating effects at detected QTL, however, where the number of loci interrogated will be fewer by several orders of magnitude and the amount of time and energy devoted to interpretation will be far greater, there is room for a different trade-off — one that emphasizes comprehensive and flexible incorporation of information and uncertainty rather than computational efficiency. Nevertheless, the comprehensive experiments demonstrated that Diploffect generates much reliable point estimations and also provides statistically valid interval estimates that are not provided by the alternative methods.

A primary motivation for developing Diploffect, and in particular to use a Bayesian approach to its estimation, is prediction: In particular, the ability to obtain for any future combination of haplotypes, covariates, and concisely-specified genetic background effects, a posterior predictive distribution for some function of the phenotype — this could be, for example, a cost or utility function whose posterior predictive distribution can inform decisions about how to prioritize subsequent experiments. Such predictive distributions are easily obtained from the MCMC procedure, and can also be extracted with only slightly more effort (via specification of $T(\boldsymbol{\theta})$ in Eq 5.17) from the Importance Sampling methods. I anticipate that, applied to (potentially multiple) independent QTL, Diploffect models will provide more robust out-of-sample predictions of phenotype in, for example, proposed crosses of multiparental recombinant inbred lines, than would be possible using ROP-based models.

6.2 Future Directions

I think that the following directions or problems need further investigations.

- **Incorporating more information in RNA-Seq quantification**

  Currently, RNA-Skim uses a relatively simple strategy to select sig-mers, which may choose sig-mers covering genomic variants such as SNP or structural variants, but this rarely happen in multiparent crosses as the samples DNA genomes in multiparent crosses are either known or relatively cheap to get as they are derived from a finite number of founders, and all founders' genome are commonly known. But the inconsistency of the DNA sequences of samples and the reference DNA genome commonly exists in other populations than multiparent crosses. If a sample carries variants differing with the reference genome, and some sig-mer covering such region is selected, RNA-Skim is not able to detect such sig-mers, thus, the quantification of the sig-mers' corresponding transcripts becomes inaccurate. Therefore, In order to generalize RNA-Skim to other populations, one possible extension for RNA-Skim is to incorporate the variant database to guide itself to select sig-mers from regions without known variants.

- **Using sig-mers for RNA-Seq differential analysis**

Determining which set of transcripts is expressed at different abundance levels in different samples is one critical step in understanding the effects of transcriptome in genetics. Some differential analysis methods (Anders and Huber, 2010b) directly use the number of reads aligned to transcripts, suffering the slow performance of the aligner and disregarding the fact that the alignments are not reliable. The notion of sig-mers seems a better alternative to the alignment in those tools, as it is much faster to count all sig-mers in RNA-Seq data than to align them to the transcriptome, providing the same level of accuracy with the alignment-dependent methods.

- **Extending HTreeQA to handle the multiple QTLs**

  The phylogeny tree must be built from the compatible region, which has no evidence for recombination. The advantage of multiparent crosses is their highly recombinant genomes, and the potential ability to model the interactions of multiple different QTLs for complex traits. Data analysts typically apply Lasso on all genome markers in order to joint estimate the effects from multiple QTLs, or find epistatic effects (interactions between two markers (Zhang et al., 2010)). On the other hand, the existing phylogeny-based approaches do not exploit such characteristic to find interesting correlations. The compatible regions of a phylogenetic tree also give us a view on ungenotyped markers, and such information is not used in the existing approaches to model multiple QTLs. It is worth exploring to build a phylogeny tree approach to detecting epistatic effects between any two phylogeny trees in the genome. Through the compatibility within the tree, we not only test any pair of markers, but also use the combination of the markers as the proxy to approximate the markers missing in the data.

- **Extending Diploffect for detecting the underlying alleles and accommodating multiple alleles**

  Diploffect assumes that the number of underlying alleles is the same with the number of founders in multiparent crosses, modeling each founder with a different effect to the phenotype. Though all haplotype effects share the same diffused prior, it still may employ a larger degree of freedom in the model when analyzing the QTL with fewer alleles. One possible way to reduce the degree of freedom in Diploffect is to put Dirichlet Process Prior on top of the haplotype effects, allowing Diploffect to do model selection and parameter estimation at the same time. The other possible solution is to use the relatedness of the local phylogeny tree built based on the haplotypes to reduce the possible number of combinations of different underlying alleles.

A primary motivation for developing Diploffect, and in particular to use a Bayesian approach to its estimation, is prediction: In particular, the ability to obtain for any future combination of haplotypes, covariates, and concisely-specified genetic background effects, a posterior predictive distribution for some function of the phenotype — this could be, for example, a cost or utility function whose posterior predictive distribution can inform decisions about how to prioritize subsequent experiments. Such predictive distributions are easily obtained from our MCMC procedure, and can also be extracted with only slightly more effort (via specification of $T(\boldsymbol{\theta})$ in Eq 5.17) from our Importance Sampling methods. I anticipate that, applied to (potentially multiple) independent QTL, Diploffect models will provide more robust out-of-sample predictions of phenotype in, for example, proposed crosses of multiparental recombinant inbred lines, than would be possible using ROP-based models.

In conclusion, I developed a series of methods for improving the efficiency and accuracy of important computational tasks in genetics, especially for conducting these tasks on multiparent crosses. The extensive experiments demonstrate that these methods can significantly improve the accuracy or save CPU running times. These methods help expedite on the computational intensive tasks and enable deep interpretation of results.

REFERENCES

Akey, J., L. Jin, and M. Xiong (2001). Haplotypes vs. single-marker linkage disequilibrium tests: what do we gain. *EUR. J. HUM. GENET. 9*, 291–300.

Anders, S. and W. Huber (2010a, October). Differential expression analysis for sequence count data. *Genome Biology 11*(10), R106.

Anders, S. and W. Huber (2010b, October). Differential expression analysis for sequence count data. *Genome Biology 11*(10), R106.

Au, K. F., H. Jiang, L. Lin, Y. Xing, and W. Wong (2010, August). Detection of splice junctions from paired-end RNA-seq data by SpliceMap. *Nucleic Acids Research 38*(14), 4570–4578.

Aylor, D. L., W. Valdar, W. Foulds-Mathes, R. J. Buus, R. A. Verdugo, et al. (2011a). Genetic analysis of complex traits in the emerging Collaborative Cross. *Genome Research 21*(8), 1213–1222.

Aylor, D. L., W. Valdar, W. Foulds-Mathes, R. J. Buus, R. A. Verdugo, et al. (2011b, August). Genetic analysis of complex traits in the emerging Collaborative Cross. *Genome Research 21*(8), 1213–1222.

Balakirev, E. S. and F. J. Ayala (2003). Pseudogenes: are they "junk" or functional DNA? *Annual review of genetics 37*, 123–151.

Barnett, D. W., E. K. Garrison, A. R. Quinlan, M. P. Strömberg, and G. T. Marth (2011, June). BamTools: a C++ API and toolkit for analyzing and managing BAM files. *Bioinformatics 27*(12), 1691–1692.

Bauman, L. E., J. S. Sinsheimer, E. M. Sobel, and K. Lange (2008, October). Mixed Effects Models for Quantitative Trait Loci Mapping With Inbred Strains. *Genetics 180*(3), 1743–1761.

Besenbacher, S., T. Mailund, and M. h. Schierup (2009). Local phylogeny mapping of quantitative traits: higher accuracy and better ranking than single-marker association in genomewide scans. *Genetics 181*(2), 747–753.

Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM 13*(7), 422–426.

Bonfert, T., G. Csaba, R. Zimmer, and C. Friedel (2012). A context-based approach to identify the most likely mapping for RNA-seq experiments. *BMC Bioinformatics 13*(Suppl 6), S9.

Broman, K. and S. Sen (2009). *A Guide to QTL Mapping with R/qtl.* New York: Springer.

Carlin, B. P. and T. A. Louis (2009). *Bayesian Methods for Data Analysis* (3 ed.).

Cavanagh, C., M. Morell, I. Mackay, and W. Powell (2008, Apr). From mutations to magic: resources for gene discovery, validation and delivery in crop plants. *Current opinion in plant biology 11*(2), 215–21.

Cheng, R., M. Abney, A. A. Palmer, and A. D. Skol (2011, July). QTLRel: an R Package for Genome-wideAssociation Studies in which Relatednessis a Concern. *BMC Genetics 12*(1), 66.

Churchill, G. a. and R. W. Doerge (1994). Empirical threshold values for quantitative trait mapping. *Genetics 138*(3), 963–971.

"Collaborative Cross Consortium" (2012). The genome architecture of the collaborative cross mouse genetic reference population. *Genetics*, 389–401.

Dadgar, A. (2013, December). Bloomd library. https://github.com/armon/bloomd.

Devlin, B. and K. Roeder (1999). Genomic control for association studies. *Biometrics 55*(4), 997–1004.

Ding, Z., T. Mailund, and Y. s. Song (2008). Efficient whole-genome association mapping using local phylogenies for unphased genotype data. *Bioinformatics 24*(19), 2215–2221.

dress, a. and m. steel (1992). Convex tree realizations of partitions. *Appl. Math. Lett. 5*(3), 3–6.

Durrant, C. and R. Mott (2010). Bayesian qtl mapping using inferred haplotypes. *Genetics*.

Ferris, M. T., D. L. Aylor, D. Bottomly, A. C. Whitmore, L. D. Aicher, T. a. Bell, B. Bradel-Tretheway, J. T. Bryan, R. J. Buus, L. E. Gralinski, B. L. Haagmans, L. McMillan, D. R. Miller, E. Rosenzweig, W. Valdar, J. Wang, G. a. Churchill, D. W. Threadgill, S. K. McWeeney, M. G. Katze, F. Pardo-Manuel de Villena, R. S. Baric, and M. T. Heise (2013, February). Modeling host genetic regulation of influenza pathogenesis in the collaborative cross. *PLoS Pathogens 9*(2), e1003196.

Flicek, P., M. R. Amode, D. Barrell, K. Beal, S. Brent, D. Carvalho-Silva, et al. (2011, December). Ensembl 2012. *Nucleic Acids Research 40*(D1), D84–D90.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software 33*(1), 1.

Fu, C.-P., V. Jojic, and L. McMillan (2014). An alignment-free regression approach for estimating allele-specific expression using rna-seq data. In *Research in Computational Molecular Biology*, pp. 69–84. Springer.

Gelman, A. and J. Hill (2007). *Data analysis using regression and multilevel/hierarchical models*, Volume Analytical methods for social research. New York: Cambridge University Press.

Google (2013, December). Protocal buffers. https://code.google.com/p/protobuf/.

Grabherr, M. G., B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, et al. (2011a, May). Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology 29*(7), 644–652.

Grabherr, M. G., B. J. Haas, M. Yassour, J. Z. Levin, D. A. Thompson, et al. (2011b, May). Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology 29*(7), 644–652.

Gregg, C., J. Zhang, B. Weissbourd, S. Luo, et al. (2010, August). High-resolution analysis of parent-of-origin allelic expression in the mouse brain. *Science 329*(5992), 643–648.

Griebel, T., B. Zacher, P. Ribeca, E. Raineri, V. Lacroix, et al. (2012). Modelling and simulating generic rna-seq experiments with the flux simulator. *Nucleic Acids Research 40*(20), 10073–10083.

Gusfield, D. (1991). An efficient algorithms for inferring evolutionary trees. *Networks 21*(1), 19–28.

Gusfield, D. (2009). The multi-state perfect phylogeny problem with missing and removable data: solutions via integer-programming and chordal graph theory. In *RECOMB*, pp. 236–252.

Guttman, M., M. Garber, J. Z. Levin, J. Donaghey, J. Robinson, et al. (2010, May). Ab initio reconstruction of cell type–specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs. *Nature Biotechnology 28*(5), 503–510.

Haley, C. and S. Knott (1992). A simple regression method for mapping quantitative trait loci in line crosses using flanking markers. *Heredity 69*(4), 315.

Harrison, P. M., D. Milburn, Z. Zhang, P. Bertone, and M. Gerstein (2003). Identification of pseudogenes in the Drosophila melanogaster genome. *Nucleic Acids Research 31*(3), 1033–1037.

Häsler, J., T. Samuelsson, and K. Strub (2007, May). Useful ‘junk’: Alu RNAs in the human transcriptome. *Cellular and Molecular Life Sciences 64*(14), 1793–1800.

Hirotsune, S., N. Yoshida, A. Chen, L. Garrett, F. Sugiyama, et al. (2003, May). An expressed pseudogene regulates the messenger-RNA stability of its homologous coding gene. *Nature 423*(6935), 91–96.

Hoerl, A. E. and R. W. Kennard (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics 12*(1), 55–67.

Hofstetter, J. R., J. a. Trofatter, K. L. Kernek, J. I. Nurnberger, and a. R. Mayeda (2003). New quantitative trait loci for the genetic variance in circadian period of locomotor activity between inbred strains of mice. *Journal of Biological Rhythms 18*(6), 450–462.

Hsieh, W. (2013, December). Stringpiece. https://chromium.googlesource.com/chromium/.

hudson, r. r. and n. l. kaplan (1985). Statistical properties of the number of recombination events in the history of a sample of dna sequences. *Genetics 111*(1), 147–164.

Hurles, M. (2004). Gene Duplication: The Genomic Trade in Spare Parts. *PLoS Biology 2*(7), e206.

Jurka, J. and T. Smith (1988, July). A fundamental division in the Alu family of repeated sequences. *Proceedings of the National Academy of Sciences of the United States of America 85*(13), 4775–4778.

Kang, H. M., N. a. Zaitlen, C. M. Wade, A. Kirby, D. Heckerman, M. J. Daly, and E. Eskin (2008). Efficient control of population structure in model organism association mapping. *Genetics 178*(3), 1709–23.

Karp, R. M. and M. O. Rabin (1987). Efficient randomized pattern-matching algorithms. *IBM Journal of Research and Development 31*(2), 249–260.

Keane, T. M., L. Goodstadt, P. Danecek, M. A. White, K. Wong, et al. (2011, September). Mouse genomic variation and its effect on phenotypes and gene regulation. *Nature 477*(7364), 289–294.

Kelada, S. N. P., D. L. Aylor, B. C. E. Peck, J. F. Ryan, U. Tavarez, R. J. Buus, D. R. Miller, E. J. Chesler, D. W. Threadgill, A. G. Churchill, F. Pardo-Manuel de Villena, and F. S. Collins (2012). Genetical analysis of hematological parameters in incipient lines of the Collaborative Cross. *G3: Genes, Genomes, Genetics 2*(2), 157–165.

Kennedy, B. W., M. Quinton, and J. A. Van Arendonk (1992). Estimation of effects of single genes on quantitative traits. *Journal of Animal Science 70*(7), 2000–2012.

Khelifi, A., K. Adel, L. Duret, D. Laurent, D. Mouchiroud, and M. Dominique (2005, January). HOPPSIGEN: a database of human and mouse processed pseudogenes. *Nucleic Acids Research 33*(Database issue), D59–66.

Kleinman, C. L. and J. Majewski (2012, March). Comment on "Widespread RNA and DNA Sequence Differences in the Human Transcriptome". *Science 335*(6074), 1302–1302.

Kover, P. X., W. Valdar, J. Trakalo, N. Scarcelli, I. M. Ehrenreich, M. D. Purugganan, C. Durrant, and R. Mott (2009, July). A Multiparent Advanced Generation Inter-Cross to Fine-Map Quantitative Traits in Arabidopsis thaliana. *PLoS Genetics 5*(7), e1000551.

Kurtz, S., A. Narechania, J. C. Stein, and D. Ware (2008). A new method to compute K-mer frequencies and its application to annotate large repetitive plant genomes. *BMC Genomics 9*(1), 517.

Lander, E. S. and D. Botstein (1989). Mapping mendelian factors underlying quantitative traits using RFLP linkage maps. *Genetics 121*(1), 185–199.

Langmead, B., C. Trapnell, M. Pop, and S. L. Salzberg (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology 10*(3), R25.

Larribe, F., S. Lessard, and N. J. Schork (2002). Gene mapping via the ancestral recombination graph. *Theoretical Population Biology 62*(2), 215 – 229.

Le, H. S., M. H. Schulz, B. M. McCauley, V. F. Hinman, and Z. Bar-Joseph (2013, May). Probabilistic error correction for RNA sequencing. *Nucleic Acids Research 41*(10), e109–e109.

Le Cam, L. (1960). An approximation theorem for the Poisson binomial distribution. *Pacific Journal of Mathematics 10*(4), 1181–1197.

Lenarcic, A., K. Svenson, G. A. Churchill, and W. Valdar (2012). A general Bayesian approach to analyzing diallel crosses of inbred strains. *Genetics 190*(2), 413–435.

Lettre, G., C. Lange, and J. N. Hirschhorn (2007). Genetic model testing and statistical power in population-based association studies of quantitative traits. *The Am. J. Hum. Genet. 362*, 358–362.

Li, B. and C. N. Dewey (2011). RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinformatics 12*, 323.

Li, B., V. Ruotti, R. M. Stewart, J. A. Thomson, and C. N. Dewey (2010, February). RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics 26*(4), 493–500.

Li, J. and T. Jiang (2005). Haplotype-based linkage disequilibrium mapping via direct data mining. *Bioinformatics 21*(2424), 4384–4393.

Li, M., I. X. Wang, Y. Li, A. Bruzel, A. L. Richards, J. M. Toung, and V. G. Cheung (2011, June). Widespread RNA and DNA Sequence Differences in the Human Transcriptome. *Science 333*(6038), 53–58.

Lin, D. Y. and D. Zeng (2006, March). Likelihood-Based Inference on Haplotype Effects in Genetic Association Studies. *Journal of the American Statistical Association 101*(473), 89–104.

Liu, E. Y., Q. Zhang, L. McMillan, F. P. M. de Villena, and W. Wang (2010, June). Efficient genome ancestry inference in complex pedigrees with inbreeding. *Bioinformatics 26*(12), i199–i207.

Liu, J. S., R. Chen, and T. Logvinenko (2001). A theoretical framework for sequential importance sampling with resampling. pp. 225–246.

Long, A. D. and C. H. Langley (1999). The power of association studies to detect the contribution of candidate genetic loci to variation in complex traits. *Genome Research 9*(8), 720–1031.

Mailund, T., S. Besenbacher, and M. H. Schierup (2006). Whole genome association mapping by incompatibilities and local perfect phylogenies. *BMC Bioinformatics 7*, 454.

Marcais, G. and C. Kingsford (2011, March). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics 27*(6), 764–770.

Mayeda, a. R. and J. R. Hofstetter (1999). A qtl for the genetic variance in free-running period and level of locomotor activity between inbred strains of mice. *Behavior Genetics 29*(3), 171–176.

McClurg, P., M. T. Pletcher, T. Wiltshire, and A. I. Su (2006). Comparative analysis of haplotype association mapping algorithms. *BMC Bioinformatics 7*, 61.

Melsted, P. and J. K. Pritchard (2011, August). Efficient counting of k-mers in DNA sequences using a bloom filter. *BMC Bioinformatics 12*(1), 333.

Minichiello, M. q. and R. Durbin (2006). Mapping trait loci by use of inferred ancestral recombination graphs. *The Am. J. Hum. Genet. 79*(5), 910–922.

Morris, A., J. Whittaker, and D. Balding (2002). Fine-scale mapping of disease loci via shattered coalescent modeling of genealogies. *The Am. J. Hum. Genet. 70*(3), 686–707.

Mott, R., C. Talbot, M. Turri, A. Collins, and J. Flint (2000). A method for fine mapping quantitative trait loci in outbred animal stocks. *Proceedings of the National Academy of Sciences of the United States of America 97*(23), 12649.

Neal, R. (2003). Slice sampling. *The Annals of Statistics*, 705–741.

Nicolae, M., S. Mangul, I. I. Măndoiu, and A. Zelikovsky (2011, April). Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms for Molecular Biology 6*(1), 9.

Onkamo, P., V. Ollikainen, P. Sevon, H. Toivonen, H. Mannila, and J. Kere (2002). Association analysis for quantitative traits by data mining: Qhpm. *Ann. Hum. Genet. 66*(5-65-6), 419–429.

Ozsolak, F. and P. M. Milos (2010, December). RNA sequencing: advances, challenges and opportunities. *Nature Publishing Group 12*(2), 87–98.

Pachter, L. (2011, April). Models for transcript quantification from RNA-Seq. *arXiv.org NA*.

Pan, F., L. McMillan, F. Pardo-Manuel De Villena, D. Threadgill, and W. Wang (2009). Treeqa: quantitative genome wide association mapping using local perfect phylogeny trees. *Pacific Symposium On Biocomputing 426*, 415–426.

Pan, F., L. Yang, L. McMillan, F. P. M. d. Villena, D. Threadgill, and W. Wang (2008). Quantitative association analysis using tree hierarchies. In *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, Washington, DC, USA, pp. 971–976. IEEE Computer Society.

Parmigiani, G. and L. Inoue (2009). *Decision theory: principles and approaches*. Chichester: Wiley.

Patro, R., S. M. Mount, and C. Kingsford (2013, August). Sailfish: Alignment-free Isoform Quantification from RNA-seq Reads using Lightweight Algorithms. *ArXiv e-prints NA*.

Pe'er, I., P. I. W. De Bakker, J. Maller, R. Yelensky, D. Altshuler, and M. J. Daly (2006). Evaluating and improving power in whole-genome association studies using fixed marker sets. *Nat. Genet. 38*(6), 663–670.

Phillippi, J., Y. Xie, D. R. Miller, T. A. Bell, Z. Zhang, A. Lenarcic, D. L. Aylor, S. H. Krovi, D. W. Threadgill, F. P.-M. de Villena, W. Wang, W. Valdar, and J. A. Frelinger (2014, November). Using the emerging Collaborative Cross to probe the immune system. *15*(1), 38–46.

Plummer, M. (2003). JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003). March*, 20–22.

Plummer, M. (2011). *rjags: Bayesian graphical models using MCMC*. R package version 3-5.

Price, A. L., N. J. Patterson, R. M. Plenge, M. E. Weinblatt, N. a. Shadick, et al. (2006). Principal components analysis corrects for stratification in genome-wide association studies. *Nat. Genet. 38*(8), 904–909.

Pruitt, K. D., T. Tatusova, and D. R. Maglott (2007, January). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Research 35*(Database), D61–D65.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

"Rat Genome Sequencing and Mapping Consortium" (2013, July). Combined sequence-based and genetic mapping analysis of complex traits in outbred rats. *Nature Publishing Group 45*(7), 767–775.

Rizk, G., D. Lavenier, and R. Chikhi (2013, February). DSK: k-mer counting with very low memory usage. *Bioinformatics 29*(5), 652–653.

Roberts, A. and L. Pachter (2013, January). Streaming fragment assignment for real-time analysis of sequencing experiments. *Nature Methods 10*(1), 71–73.

Robertson, G., J. Schein, R. Chiu, R. Corbett, M. Field, et al. (2010, October). De novo assembly and analysis of RNA-seq data. *Nature Methods 7*(11), 909–912.

Ronnegard, L. and W. Valdar (2011a, June). Detecting Major Genetic Loci Controlling Phenotypic Variability in Experimental Crosses. *Genetics 188*(2), 435–447.

Ronnegard, L. and W. Valdar (2011b, June). Detecting Major Genetic Loci Controlling Phenotypic Variability in Experimental Crosses. *Genetics 188*(2), 435–447.

Rue, H., S. Martino, and N. Chopin (2009). Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the royal statistical society: Series b (statistical methodology) 71*(2), 319–392.

Scheet, P. and M. Stephens (2006). A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *Am. J. Hum. Genet. 78*(4), 629–644.

Sevon, P., H. Toivonen, and V. Ollikainen (2006). Treedt: tree pattern mining for gene mapping. *IEEE/ACM Trans. Comput. Biol. Bioinf. 3*(2), 174–1085.

Sillanpaa, M. J. and E. Arjas (1998, March). Bayesian mapping of multiple quantitative trait loci from incomplete inbred line cross data. *Genetics 148*(3), 1373–1388.

Sillanpaa, M. J. and E. Arjas (1999, March). Bayesian mapping of multiple quantitative trait loci from incomplete outbred offspring data. *Genetics 151*(4), 1605–1619.

Solberg, L. C., A. E. Baum, N. Ahmadiyeh, K. Shimomura, R. Li, F. W. Turek, J. S. Takahashi, G. A. Churchill, and E. E. Redei (2006, July). Genetic analysis of the stress-responsive adrenocortical axis. *Physiological Genomics 27*(3), 362–369.

Svenson, K., D. Gatti, W. Valdar, C. Welsh, R. Cheng, E. J. Chesler, A. Palmer, L. McMillan, and G. A. Churchill (2012). High-resolution genetic mapping using the mouse diversity outbred population. *Genetics 190*(2), 437–447.

Thomas, D. C. (2004). *Statistical methods in genetic epidemiology*. Oxford University Press, USA.

Trapnell, C., L. Pachter, and S. L. Salzberg (2009, May). TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics 25*(9), 1105–1111.

Trapnell, C., A. Roberts, L. Goff, G. Pertea, D. Kim, et al. (2012a, March). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature protocols 7*(3), 562–578.

Trapnell, C., A. Roberts, L. Goff, G. Pertea, D. Kim, et al. (2012b, March). Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks. *Nature protocols 7*(3), 562–578.

Trapnell, C., B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, et al. (2010, May). Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology 28*(5), 516–520.

Turro, E., S.-Y. Su, Â. Gonçalves, L. J. Coin, S. Richardson, and A. Lewin (2011, February). Haplotype and isoform specific expression estimation using multi-mapping RNA-seq reads. *Genome Biology 12*(2), R13.

Uziela, K. and A. Honkela (2013, April). Probe region expression estimation for RNA-seq data for improved microarray comparability. *ArXiv e-prints NA*.

Valdar, W., J. Flint, and R. Mott (2006). Simulating the collaborative cross: power of quantitative trait loci detection and mapping resolution in large sets of recombinant inbred strains of mice. *Genetics 172*(3), 1783–1797.

Valdar, W., C. C. Holmes, R. Mott, and J. Flint (2009, August). Mapping in Structured Populations by Resample Model Averaging. *Genetics 182*(4), 1263–1277.

Valdar, W., L. C. Solberg, D. Gauguier, S. Burnett, P. Klenerman, W. O. Cookson, M. S. Taylor, J. N. P. Rawlins, R. Mott, and J. Flint (2006, July). Genome-wide genetic association of complex traits in heterogeneous stock mice. *Nature Genetics 38*(8), 879–887.

van Bakel, H., C. Nislow, B. J. Blencowe, and T. R. Hughes (2010, May). Most "Dark Matter" Transcripts Are Associated With Known Genes. *PLoS Biology 8*(5), e1000371.

Vanin, E. F. (1985). Processed pseudogenes: characteristics and evolution. *Annual review of genetics 19*, 253–272.

Vazquez, A. I., D. M. Bates, G. J. M. Rosa, D. Gianola, and K. A. Weigel (2010, January). Technical note: An R package for fitting generalized linear mixed models in animal breeding. *Journal of Animal Science 88*(2), 497–504.

Wang, K. and V. Sheffield (2005). A constrained-likelihood approach to marker-trait association studies. *The Am. J. Hum. Genet. 77*(5), 768–780.

Wang, K., D. Singh, Z. Zeng, S. J. Coleman, Y. Huang, et al. (2010, October). MapSplice: Accurate mapping of RNA-seq reads for splice junction discovery. *Nucleic Acids Research 38*(18), e178–e178.

Wang, Z., M. Gerstein, and M. Snyder (2009, January). RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews. Genetics 10*(1), 57–63.

Westfall, P. H. and S. S. Young (1993). *Resampling-based multiple testing : examples and methods for P-value adjustment*. Wiley.

Woods, L. C. S., K. L. Holl, D. Oreper, Y. Xie, S. W. Tsaih, and W. Valdar (2012, November). Fine-mapping diabetes-related traits, including insulin resistance, in heterogeneous stock rats. *Physiological Genomics 44*(21), 1013–1026.

Xing, Y., T. Yu, Y. N. N. Wu, M. Roy, J. Kim, and C. Lee (2006). An expectation-maximization algorithm for probabilistic reconstructions of full-length isoforms from splice graphs. *Nucleic acids research 34*(10), 3150–3160.

Yang, H., Y. Ding, L. Hutchins, J. Szatkiewicz, T. Bell, et al. (2009). A customized and versatile high-density genotyping array for the mouse. *Nat. Meth. 6*(9), 663–666.

Yang, H., J. R. Wang, J. P. Didion, R. J. Buus, T. a. Bell, et al. (2011). Subspecific origin and haplotype diversity in the laboratory mouse. *Nat. Genet. 43*(7), 648–655.

Zhang, X., S. Huang, F. Zou, and W. Wang (2010, June). TEAM: efficient two-locus epistasis tests in human genome-wide association study. *Bioinformatics 26*(12), i217–i227.

Zhang, Z., P. M. Harrison, Y. Liu, and M. Gerstein (2003, December). Millions of years of evolution preserved: a comprehensive catalog of the processed pseudogenes in the human genome. *Genome Research 13*(12), 2541–2558.

Zhang, Z., S. Huang, J. Wang, X. Zhang, F. P.-M. de Villena, et al. (2013). GeneScissors: a comprehensive approach to detecting and correcting spurious transcriptome inference owing to RNA-seq reads misalignment. *Bioinformatics 29*(13), 291–299.

Zhang, Z. and W. Wang (2014). Rna-skim: a rapid method for rna-seq quantification at transcript-level. *Bioinformatics*, to appear.

Zhang, Z., W. Wang, and W. Valdar (2014). Bayesian modeling of haplotype effects in multiparent populations. *Genetics*, in submission.

Zhang, Z., X. Zhang, and W. Wang (2012). HTreeQA: Using semi-perfect phylogeny trees in quantitative trait loci study on genotype data. *G3: Genes| Genomes| Genetics 2*(2), 175–189.

Zöllner, S. and J. K. Pritchard (2005). Coalescent-based association mapping and fine mapping of complex trait loci. *Genetics 169*(2), 1071–1092.