**Title**
Probabilistic Methods for Single Individual Haplotype Reconstruction: HapTree and HapTree-X



**Author**
Berger, Emily Rita

**Publication Date**
2015

Peer reviewed|Thesis/dissertation

**Probabilistic Methods for Single Individual Haplotype Reconstruction: HapTree and HapTree-X**

by

Emily Rita Berger

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Mathematics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Lior Pachter, Chair
Professor Steven N. Evans
Associate Professor Yun S. Song

Summer 2015

**Probabilistic Methods for Single Individual Haplotype Reconstruction: HapTree and HapTree-X**

Abstract

Probabilistic Methods for Single Individual Haplotype Reconstruction:
HapTree and HapTree-X

by

Emily Rita Berger

Doctor of Philosophy in Mathematics

University of California, Berkeley

Professor Lior Pachter, Chair

Identifying phase information is biomedically important due to the association of complex haplotype effects, such as compound heterozygosity, with disease. As recent next-generation sequencing (NGS) technologies provide more read sequences, the use of diverse sequencing datasets for haplotype phasing is now possible, allowing haplotype reconstruction of a single sequenced individual using NGS data. Nearly all previous haplotype reconstruction studies have focused on diploid genomes and are rarely scalable to genomes with higher ploidy. Yet computational investigations into polyploid genomes carry great importance, impacting plant, yeast and fish genomics, as well as the studies of the evolution of modern-day eukaryotes and (epi)genetic interactions between copies of genes. Furthermore, previous diploid haplotype reconstruction studies have ignored differential allele-specific expression in whole transcriptome sequencing (RNA-seq) data; however, intuition suggests that the asymmetry in this data (i.e. maternal and paternal haplotypes of a gene are differentially expressed) can be exploited to improve phasing power. In this thesis, we describe novel integrative maximum-likelihood estimation frameworks, HapTree and HapTree-X, for efficient, scalable haplotype assembly from NGS data. HapTree is built to recover an individual polyploid genome from genomic read data, and HapTree-X aims to reconstruct a diploid genome or transcriptome from RNA-seq and DNA-seq data by making use of differential allele-specific expression. HapTree-X is the first method for haplotype assembly that uses differential expression, newly allowing the use of reads that cover only one SNP.

For triploid and higher ploidy genomes, we demonstrate that HapTree substantially improves haplotype assembly accuracy and efficiency over the state-of-the-art; moreover, HapTree is the first scalable polyplotyping method for higher ploidy. As a proof of concept, we also test our method on real sequencing data from NA12878 (1000 Genomes Project) and evaluate the quality of assembled haplotypes with respect to trio-based diplotype annotation as the ground truth. The results indicate that HapTree significantly improves the switch accuracy within phased haplotype blocks as compared to existing haplotype assembly methods, while producing comparable minimum error correction (MEC) values. We evaluate the per-

formance of HapTree-X on real sequencing read data, both transcriptomic and genomic, from NA12878 (1000 Genomes Project and Gencode) and demonstrate that HapTree-X increases the number of SNPs that can be phased and sizes of phased-haplotype blocks, without compromising accuracy. We prove theoretical bounds on the precise improvement of accuracy as a function of coverage which can be achieved from differential expression-based methods alone. Thus, the advantage of our integrative approach substantially grows as the amount of RNA-seq data increases.

To James McClave

For always believing in me.

# Contents

# List of Figures

# List of Tables

## Acknowledgments

I would to thank my advisor, Lior Pachter, for giving me so much support and freedom, without which I could not have completed my PhD. I am also grateful to the other members of my committee, Yun Song and Steve Evans, for their support and valuable feedback on my thesis. I wish to express my great appreciation to Martin Olsson, Barb Waller, and the department for granting me the substantial flexibility that was critical to my earning this degree.

I especially must thank my mentor and aunt, Bonnie Berger, for being my role model, continually encouraging and inspiring me, and never letting me quit. Your faith in me over the last decade has been essential — thank you Bonnie. Bonnie also made it possible for me to work in her lab at MIT, where I've had the privilege to get to know Jian Peng, Noah Daniels, George Tucker, Sean Simmons, Deniz Yorukoglu, and William Yu. A special thanks in particular to Deniz for introducing me to the field of computational biology and facilitating this thesis.

While the work does not appear in this thesis, a year of my research in graduate school was supervised by Alistair Sinclair and Peter Shor; from them I learned so much and am very grateful for their guidance and teaching. Furthermore, my favorite of the courses I took throughout graduate school was one taught by Alistair Sinclair my first year; I was very inspired and motivated by this class and wish to thank Alistair for providing me with this experience.

I'd like to acknowledge all of my colleagues from both Berkeley and MIT; I have been so inspired by their brilliance, dedication, and passion. Thanks in particular to my office mates Madars Virza, Alexander Dubbs, Ludwig Schmidt, and William Leiserson at MIT and Gus Schrader at Berkeley for sharing your days with me! A special thanks to Madars for seeming to always share my deadlines.

Finally, I want to thank my family and friends; I would not be here without all of you. I must thank my dog, Mia, for patiently sitting with me for ever hour I spent writing this thesis. Lastly, I especially want to thank my grandfather for introducing me to the beauty of mathematics.

# Chapter 1

# Introduction

The human genome has been successfully sequenced using whole genome shotgun sequencing technology, creating a human reference genome. The average human's genome agrees with the reference genome in approximately 99.5 % of locations. The variation between genomes accounts for the biological differences between individuals, and thus much important genetic information is not contained within the reference genome. A main goal of computational genomics is to understand these differences and their impact.

Much variation is contained within the genotypes of an individual at single nucleotide polymorphism (SNP) sites. Methods for genotyping, or determining the set of alleles inherited from an individual's parents at a particular locus, have been around the decades. The *haplotypes* of an individual are the sequences of these alleles, indicating on which homologous chromosome each allele falls, and thus contain more information than the genotypes alone.

Determining haplotypes is biologically and computationally more difficult than determining the genotypes alone. The problem of single individual haplotype assembly is to determine the haplotypes of an individual using sequenced reads (short pieces of his or her genome or transcriptome.) This thesis introduces novel probabilistic algorithms for determining haplotype blocks in several different biological contexts.

## 1.1 Diploid Haplotypes

Humans are diploid organisms, having two copies of each chromosome (other than the sex chromosomes), or two *haplotypes*. In almost all locations, the two haplotypes belonging to an individual are identical. The locations in the genome where the haplotypes can commonly vary throughout the population correspond to the "single nucleotide polymorphisms", or SNPs. For any individual, SNP sites where the same allele occurs on each haplotype are called homozygous SNPs, and SNP sites where different alleles occur on each haplotype are called heterozygous SNPs. In the case of heterozygous SNPs, we call one allele the "wild" type or reference allele and one allele the "mutant" or alternative allele.

## 1.2 Polyploid Haplotypes

While humans are diploid and have two copies of each chromosome, this is not the case for all organisms. Unlike humans, plants and certain fish (among other organisms) may have more than two copies of each chromosome; these species are considered to be *polyploid*. For example, salmon, goldfish, and salamanders are polyploid; wheat, depending on the strain, can be diploid, tetraploid (4 copies) or hexaploid (6 copies). More generally, we say a species is $k$-ploid if it has $k$ copies of each homologous chromosome.

Throughout this thesis, we assume that the ploidy is always known, but this is not necessarily the case in general. Determining ploidy is a complex problem in its own right and shares many features with several problems in the field of metagenomics. Another complexity that can occur in relation to polyploid genomes is non-constant ploidy across a genome: that is, the ploidy varies throughout the genome, such as in tumor cells or even in full chromosomes. The latter is the case for Down syndrome for example, which is caused by an individual having three copies (triploid) chromosome 21.

## 1.3 RNA vs DNA

While DNA and RNA are both chains of nucleotides, their functions and structure are distinct. Normal cellular function consists of replication of DNA, translation of DNA into RNA, and transcription of RNA into proteins. DNA once translated becomes messenger RNA (mRNA) and is spliced into exons (coding regions) and introns (non-coding regions). It is possible for there to be multiple splicings of RNA into exons; we refer to these splicings as isoforms. Unlike DNA, RNA is usually single stranded and the two strands may be expressed differently. In this thesis, we design an algorithm, HapTree-X, which leverages this differential expression to preform phasing.

## 1.4 Related Problems and Methods

Various sources of information can be utilized for the computational identification of an individual's diplotype/polyplotype: pedigree (e.g. trio-based phasing) [31, 32, 8], population structure of variants (e.g. phasing by linkage disequilibrium) [8, 30, 29, 12] and more recently by identity-by-descent in unrelated individuals [9, 4], as well as sequencing read datasets [5, 3, 18, 6, 14]. Among these approaches, methods for sequence-based haplotype phasing are the only viable approach for haplotype phasing on a single individual member of a species (assuming homologous chromosomes are sequenced together), as other approaches either require family members or a population.

For an individual diploid genome, the problem of reconstructing the diplotype using sequence information, the diploid phasing problem, is equivalent to the identification of the sequence of alleles on either parental haplotype. If this sequence is correctly inferred, then the other haplotype will automatically carry the corresponding opposite alleles (reference or

alternative). Solving an error-free version of the diploid haplotype reconstruction problem is straightforward: the haplotype of each connected (by reads) component of heterozygous SNPs can be obtained by propagating allele information within reads. In reality, however, sequencing errors as well as false read mappings cause conflicts within sequence information, requiring a mathematical formulation of the haplotype reconstruction problem.

Among various formulations suggested for this problem, the most commonly used is an NP-hard minimum error correction (MEC) definition [20, 22], which aims to identify the smallest set of nucleotide changes required within mapped fragments that would allow a conflict-free separation of reads into two separate homologous chromosomes (or a bipartite separation of the fragment conflict graph). Some of the solutions proposed for this problem include: HapCUT [5], an algorithm for optimizing MEC score based on computing max-cuts of the fragment graph; Fast Hare [24], a heuristic that clusters reads into two sets in a greedy fashion, and HapCompass [3], a spanning tree based approach for minimizing fragment conflicts.

## 1.5 Motivation

While knowing the haplotypes of an individual has value in its own right, there are also specific applications from having this information. For example, biomedical studies that focus on certain autosomal recessive disorders must determine whether or not a given gene is a compound heterozygote. In order to do so, the individual's haplotypes, in addition to the genotypes, must be known.

### Compound Heterozygosity

By running standard genotype calling tools, it is possible to accurately identify the number of "wild type" and "mutant" alleles (A, C, G, or T) for each single-nucleotide polymorphism (SNP) site. However, in the case of two heterozygous SNP sites, genotype calling tools cannot determine whether "mutant" alleles from different SNP loci are on the same or different chromosomes (i.e. compound heterozygote). While the former would be healthy, in many cases the latter can cause loss of function; it is therefore necessary to identify the phase (*phasing*) — the copies of a chromosome on which the mutant alleles occur — in addition to the genotype (Figure 1.1).


**Figure 1.1:** Loss of function in different polyplotypes of a sample pentaploid genome

As the loss of function is often determined by whether a healthy copy of a gene exists, knowing the genotype vector is sufficient if there is a single SNP site. In the case of two SNP sites however, the genotype vector cannot be used to unambiguously determine loss of function, and phasing is required.

## 1.6 Single Individual Haplotyping: The Problem

The problem of single individual haplotyping is to determine the haplotypes of an individual from next-generation sequencing (NGS) data. This data contains contiguous DNA (or RNA) segments, for which a mapper may then be used to map these fragments to a reference genome. While these fragments have been mapped to locations within the genome, we have no information regarding which homologous chromosome they were sequenced from, except that each fragment is derived from a single haplotype. A useful fragment is therefore one which covers at least two heterozygous SNPs, and thus contains information about the phase of the haplotype from which it was sequenced.

Approaches to haplotype assembly focus on reconstructing the haplotypes from sequenced fragments which cover more than one heterozygous SNP to determine which alleles fall on which homologous chromosomes. Due to the large distances which can occur between SNPs, the entire haplotypes cannot actually be determined, but rather pieces of the genome can be phased; we refer to these pieces as *phased haplotype blocks*. For details and the formal definition of a phased haplotype block, she section 2.1.

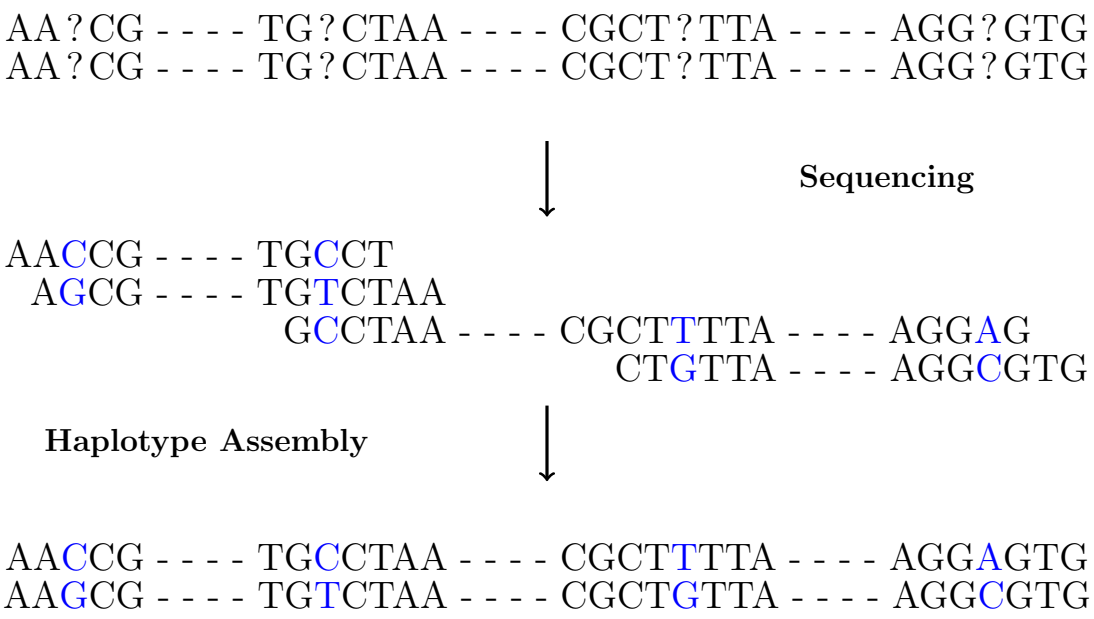


**Figure 1.2:** Sequence and Haplotype Assembly

## Diploid

Methods for phasing a diploid genome in the past have generally aimed to minimize the number of conflicts between sequenced reads; any two reads conflict if once assigned to a particular haplotype they disagree at an overlapping SNP locus. If there were no errors in reads, it would be trivial to find a solution with no conflicts. Unfortunately reads are not error free, and in the case of gapped reads minimizing the number of conflicts, or minimum error correction (MEC), is known to be computationally intractable. We take a different approach, instead attempting to find a solution of maximum likelihood, as opposed to minimum conflict. We remark that if we assume uniform read errors and that each read covers only two SNPs, a solution of maximum likelihood is one of minimum conflict, and vice versa.

## Polyploid Variation

Unlike diploid genomes, computational identification of common chromosomal variants in polyploid genomes using sequencing data has received little attention, except in the pioneering work of Aguiar & Istrail [4]. Polyploidy studies are of importance as they allow a comprehensive investigation of variants within plant, fish, and yeast genomes and help understand mechanisms of eukaryotic evolution. However, haplotype reconstruction in polyploid genomes is fundamentally more complex, even in the error-free version of the problem (without sequencing errors or false read mappings). Due to the newness of the NGS-based biological research in polyploid genomes, the mathematical foundations of the polyploid

phasing problem have not yet been established. The solution proposed by Aguiar & Istrail for single individual polyplotyping problem is based on phasing all possible SNP loci pairs independently while further consolidating this information in a separate stage in order to infer a set of haplotypes.

Diploid phasing methods focus on a given list of heterozygous variants that are guaranteed to contain a single reference allele, as well as an alternative allele (even assuming the simpler case in which all heterozygous loci are bi-allelic). In contrast, in the polyploid phasing problem, there is no such guarantee of a single type of heterozygous SNP. In the simplest case where all SNP loci are bi-allelic, each heterozygous locus for a $k$-ploid chromosome can potentially contain from 1 up to $k-1$ alternative alleles within the heterozygous loci, significantly increasing the complexity of the phasing problem in comparison to the diploid case. Furthermore, in a diploid phasing setting, there are always two possible options for phasing a pair of SNP loci, regardless of what other SNPs they are phased with. These two options can be thought as parallel (alternative allele pairs and reference allele pairs are matched within themselves) or switched (each alternative allele is matched with the other reference allele). These two options are no longer sufficient when the genome contains more than two copies of each chromosome, due to the fact that there are up to $k!$ options when merging a phased haplotype block with another.

## RNA-seq variation

There are two major differences between RNA-seq data and DNA-seq data which may both be leveraged to produce a more powerful algorithm for haplotype reconstruction. Firstly, in the case of DNA-seq data, both the maternal and paternal haplotypes are expressed equally; this is not necessarily the case in RNA-seq data. RNA-seq data features *differential haplotypic expression*, or DHE: the maternal and paternal haplotypes are not necessarily expressed at the same rate. DHE in the transcriptome can be exploited to improve phasing power because SNP alleles within maternal and paternal haplotypes of a gene are present in the read data at (different) frequencies corresponding to the differential haplotypic expression (DHE).

This asymmetry in the data due to differential haplotypic expression not only contains a significant amount of information, but furthermore allows us to make use of reads which cover only one heterozygous SNP. To date, approaches that utilize RNA-seq data for phasing (or DNA-seq data) (e.g., [25]) can only make use of reads covering 2 or more heterozygous SNPs, as they repurpose existing genome phasing approaches which are based on sequence contiguity. However, only 10% of reads that overlap a heterozygous SNP fall into this category (Table 5.2). Thus, current methods are discarding 90% of potentially useful information. Though these reads do not overlap multiple SNPs, as do those conventionally used for phasing, they provide insight into differential haplotypic expression within genes. An advantage of using reads covering only a single SNP is that phasing is not limited by the length of the read or fragment, nor the transcriptomic or genomic distance between SNPs.

A second important feature of RNA-seq data, is that compared to DNA-seq, RNA-seq allows for longer-range phasing due to RNA splicing in the transcriptome. While the length of the fragment is the same in RNA-seq experiments as in DNA-seq experiments, the fragments only cover exons, and therefore connected SNPs much further apart in the genome that previously possible. This ultimately leads to larger phased blocks and more SNPs that can be phased.

## 1.7 Data

HapTree and HapTree-X were designed to perform phasing of a genome or transcriptome using Next-Generation Sequencing (NGS) data. Several NGS methods have been developed including but not limited to Sanger sequencing, 454, and Illumina. These methods vary both in what is performed at a biological level and what sort of data is generated. In all cases, a DNA (or RNA) sample is broken up into small fragments which can be sequenced, generating millions of *reads* consisting of a (possibly paired-end) sequence of nucleotides. The length of the read (total number of nucleotides sequenced) as well as the rates of error (probability of reporting an incorrect nucleotide) vary across methods, as do the costs per base pair sequenced.

In the case of genomic sequencing data (DNA-seq), haplotypes are expressed equally, and each read is theoretically equally likely to have come from any haplotype. In the case of transcriptomic sequencing data (RNA-seq) however, genes may be differentially expressed. Furthermore, because of the exon/intron spliced structure of RNA, RNA-seq reads can cover pairs of SNPs at much greater genomic distance than DNA-seq reads of the same length and insert size because the RNA-seq read is restricted to the exons in the transcriptome.

To run HapTree or HapTree-X, the following data types are required:

### Reference Genome

The reference genome must be in the .fa format and contains a guess about the information of the common reference alleles for an individual in the population.

### Variant Calls File (VCF)

The VCF file contains all the necessary information about the homozygous and heterozygous SNP sites and the reference and alternative alleles for each SNP. For polyploid genomes, the genotypes are extremely useful to have within the VCF, though not necessary.

### BAM/SAM Format

A BAM or SAM file contains NGS read data. To use with HapTree(-X), the BAM or SAM file ought to contain uniquely mapped reads and we supply a tool, Chair, to translate the SAM file into an easy to read list of reads restricted only to the heterozygous SNP sites which they cover. Chair takes as input a VCF file, reference genome, and SAM file.

### Gene Model (BED Format)

To use HapTree-X, a gene model must be assumed in order to determine what SNPs blocks may be phased and where DHE ought to be constant. HapTree-X uses a BED format as input, though the different formats for gene models are easily translated into one another.

## 1.8 Overview

This thesis has been broken down into six chapters, including this introduction. The remaining chapters are as follows.

### Chapter 2: Relative Likelihood Models: HapTree

In this chapter we build the theoretical framework for HapTree, including a relatively likelihood score which will be the metric that HapTree aims to maximize throughout. This likelihood score results from a model we derive for the probability that a haplotype is the true haplotype given a read dataset. We apply Bayes' theorem and condition instead on the haplotype in question and attempt to model the probability of generating the read dataset given a particular haplotype. Furthermore, we discuss how to this model can be extended in a natural way to handle the cases of multi-allelic SNPs, unknown genotypes, and partially known haplotypes.

### Chapter 3: Relative Likelihood Model: HapTree-X

We generalize the theory from Chapter 2 to handle the case of RNA-seq data. In this context, the main difference between RNA-seq and DNA-seq is that in RNA-seq data, maternal and paternal haplotypes may be differentially expressed. The HapTree model assumes all reads are equally likely to have come from the maternal or paternal haplotype (in the diploid case); we generalize that model to accommodate this non-uniformity.

We demonstrate how to estimate the differential expression for each gene by formulating a Hidden Markov Model and applying the forward algorithm to find the maximum likelihood rate of differential expression. These estimates are to determine which genes we believe have sufficient differential expression to be phased using this method. If a gene is found to have sufficient differential expression, this estimate is then used as input for the relative likelihood function.

Finally, we define the event *concordant expression* which measures whether differential expression is occurring as expected. We compute the probability of concordant expression as well as show that under very mild assumptions, the solution of maximal likelihood is that with concordant expression everywhere.

## Chapter 4: Maximization of Relative Likelihood Score

We aim to maximize both the relative likelihood models for HapTree and HapTree-X in the same way. In the most general case, to provably find the solution of maximal likelihood, one would need to enumerate all possible haplotype blocks or use exact dynamic programming, which for some cases is intractable. Therefore, we develop an algorithm which in practice finds solutions of high likelihood quickly. Our solution is based on finding high likelihood phases for the first $m+1$ SNPs, conditioned on a collection of high likelihood phases for the first $m$ SNPs.

## Chapter 5: Simulated and Experimental Results

To determine the accuracy of HapTree and HapTree-X, we tested both on simulated and experimental data. For the case of HapTree, we compared against HapCut [5] on NA12878 (1000 Genomes Project) data and found that HapTree slightly outperformed HapCut both in accuracy and speed. HapTree was designed specially for the polyploid case; we simulated polyploid data and compared HapTree to HapCompass [3, 4] and found that HapTree drastically outperformed HapCompass. We show that MEC score, which was previously used in many diploid phasing algorithms [5], is not sensitive enough of a metric to be used for polyploid phasing.

For HapTree-X, we demonstrate that as coverage increases, so does the theoretical accuracy. Because HapTree-X does not require all useable reads to cover at least two SNPs, we are able to increase the number of SNPs phased as well as the lengths of the phased blocks. We compared HapTree-X to HapCut to find that not only were the total SNPs phased and block lengths increased as expected, but that phasing accuracy increased substantially as well. Furthermore, HapTree-X takes in several parameters to determine how stringent to be with respect to which SNPs it elects to phase; we show that accuracy behaves as expected as a function of these parameters.

## Chapter 6: Phasing with Multiple Isoforms

In Chapter 3 of this thesis, to perform phasing based on the differential expression of a gene, we restrict to phasing subsets of the SNPs in the gene such that each SNP is covered by the same set of isoforms. This condition is trivially satisfied when a gene has only one isoform, but fails to be satisfied for many non-trivial (size greater than one) subsets when a gene has multiple isoforms.

In this chapter, we investigate how one can attempt to reconstruct haplotypes in the presence of multiple isoforms. We begin by modeling the proportions of reference and alternative alleles (at the SNP loci of a particular gene) expected to be observed in an idealized RNA-seq dataset. For a given gene, assuming the structure of its isoforms and their quantifications are known, we derive conditions describing when a particular haplotype is a *feasible haplotype* given an observation of allele proportions. A haplotype is a feasible haplotype when it is (mathematically) possible for it to have been the underlying haplotype given the observation of allele proportions.

We extend this model to incorporate noisy differential expression and investigate (for GM12878 transcriptome fragments sequenced from the nucleus) how often there is a unique feasible solution, and when that solution is correct with respect to the platinum phased Illumina vcf file [1]. We find that at this time, certain SNP pairs can be very accurately phased in the presence of multiple isoforms: those which satisfy various error and coverage thresholds. Unfortunately, a small percentage of all SNP pairs satisfy those constraints.

# Chapter 2

# Relative Likelihood Models: HapTree

## 2.1 Definitions and Notation

We describe below the problem of sequence-based single individual haplotype assembly for polyploid (and diploid) genomes and provide basic technical notation that will be useful for describing our method.

### Genotypes

We assume for now that each SNP locus to be phased is bi-allelic (i.e. contains only two possible alleles, one being the reference allele). We further assume that for each SNP locus $s$, the genotype of $s$ is known and is defined to be the number of chromosomes carrying the alternative allele (denoted by $g(s)$). For diploid haplotypes, the genotype will be 0 for homozygous SNPs (which need not to be phased as the same allele occurs on both copies on the homologous chromosomes) and 1 for heterozygous SNPs (there is always one reference allele and one alternative allele at each locus). In the polyploid case, let $k$ denote the ploidy, then $g(s)$ can range from 1 to $k-1$ for heterozygous loci $s$. We note that these two assumptions are made for the sake of simplicity in describing our model. At the end of this chapter we discuss how to update our model to the general case.

### Reads

We denote the sequence of observed nucleotides of a fragment simply as a "read" (independent from single/paired-end reads and sub-reads of a strobe read structure). The set of all reads is denoted as $R$. We define a read $r \in R$ as a vector with entries $r[i] \in \{0, 1, -\}$ where a 0 denotes the reference allele, a 1 the alternative allele, and a $-$ indicates one of two possibilities: First, that the read does not overlap with the corresponding SNP locus, or second, that neither the reference nor alternative allele is present and hence there must be a read error. A read $r \in R$ *contains* a SNP $s$ if $r[s] \neq -$. A read can also be represented as a dictionary or mapping with keys the positions (from amongst the SNPs to be phased) of

SNP loci it contains and values of either reference allele or alternative allele, represented by 0 and 1 respectively (e.g. $r = \{3{:}0,\ 4{:}1,\ 5{:}0,\ 8{:}1,\ 9{:}1\}$).

## Read Errors

As current sequencing technologies generate read data with a certain rate of sequencing errors, some of the positions within a read likely contain false nucleotide information. Among these erroneous bases, unless they are located at SNP loci and contain opposite allele information, we ignore them by representing them with $-$, and thus keep only confounding sequencing errors that can affect phased haplotype results. For each read $r$ and for each SNP locus $s$, we assume an error rate of $\epsilon_{r,s}$ and a probability of opposite false allele information $r[s]$ is equal to $\varepsilon_{r,s} = \frac{\epsilon_{r,s}}{1-\frac{2}{3}\epsilon_{r,s}}$. We modify this error rate by a factor of two-thirds because conditional on there being an error, we model the error as equally likely to be any of the three other alleles. Two of the three of these alleles are neither the reference nor the alternative allele and thus we know that an error has been made in this case. Therefore, two-thirds of the time the erroneous alleles produced are known as such and may be thrown out, leaving a true error only one-third of the time. We represent these error rates as matrices $\epsilon, \varepsilon$. At this time our method assumes uniform error rates with respect to the SNP position; the error rate is supplied by the user and ought to depend on the read sequencing technologies used.

## Read Graph

Upon the set of SNP loci $S$ and read set $R$; we define a *Read Graph*, $G(S, R)$, such that there is a vertex for each SNP locus $s \in S$ and an edge between any two vertices $s_1, s_2$ if there is some read containing both $s_1$ and $s_2$; equivalently if $\exists r \in R, r[s_1] \neq - \wedge r[s_2] \neq -$. Without loss of generality, we assume that $G(S, R)$ is connected; otherwise each connected component can be processed independently. We will define a more complex Read Graph structure in section 3.1, when we discuss the more general model that handles RNA-seq data. Below in Figure 2.1, we provide a sample set of five reads $R = \{r_1, ..., r_5\}$ on $S = \{v_1, ..., v_7\}$; the graph $G(S, R)$ has two connected components, SNPs $v_1, ..., v_5$ and $v_6, v_7$.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $r_1$ | 1 | 1 | – | – | – | – | – |
| $r_2$ | 1 | – | 0 | – | – | – | – |
| $r_3$ | – | – | 1 | 0 | 0 | – | – |
| $r_4$ | – | – | – | – | – | 1 | 0 |
| $r_5$ | – | – | – | – | – | 0 | 1 |

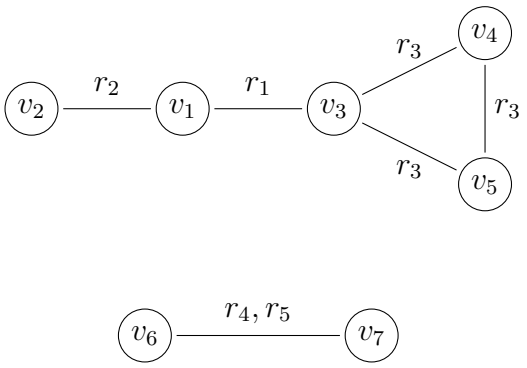


**Figure 2.1:** Example reads and corresponding Read Graph $G$

## Vector Set

A *k-ploidy phase* of $n$ SNPs with genotypes $\{g(s)\}$ is a tuple of $k$ vectors (not necessarily distinct) $(h_1, ..., h_k) \subset \{0,1\}^n$ satisfying the genotype allele counts property, that is: $h_1[s] + h_2[s] + ... + h_k[s] = g(s)$ for all $s \in \{1, 2, ..., n\}$. We will refer to this collection as a *vector set* and we think of each vector as a row vector; these vectors correspond to haplotypes.

We can build a phase by selecting a permutation of the alleles present for each SNP locus $s$. Note that the number of distinct permutations, $C(s)$, is strictly dependent on the genotype of the SNP and in the diploid bi-allelic case is equivalent to selecting the chromosomes containing the alternative alleles, hence

$$C(s) = \binom{k}{g(s)} = \frac{k!}{g(s)!(k - g(s))!}.$$

For example, let $k = 4$, then $g(s) \in \{1, 2, 3\}$. We enumerate the possible permutations below in Figure 2.2 and include an example tetraploid genome as well.

| | | |
|---|---|---|
| 0 0 0 1 | 0 0 0 1 1 1 | 0 1 1 1 |
| 0 0 1 0 | 0 1 1 0 0 1 | 1 0 1 1 |
| 0 1 0 0 | 1 0 1 0 1 0 | 1 1 0 1 |
| 1 0 0 0 | 1 1 0 1 0 0 | 1 1 1 0 |

**Figure 2.2:** All possible permutations of tetraploid bi-allelic heterozygous SNPs.

The sample tetraploid genome featured below has genotype vector: $[2, 1, 2, 3, 2, 2, 3, 2]$ (see Figure 2.3); recall this counts the number of alternative alleles present at each SNP site.

$$\begin{array}{c} 0\ 0\ 0\ 1\ 1\ 0\ 1\ 0 \\ 0\ 1\ 1\ 1\ 0\ 0\ 1\ 1 \\ 1\ 0\ 1\ 1\ 1\ 1\ 0\ 0 \\ 1\ 0\ 0\ 0\ 0\ 1\ 1\ 1 \end{array}$$

$$\{(00011010),\ (01110011),\ (10111100),\ (10000111)\}$$

**Figure 2.3:** A sample tetraploid genome and its corresponding vector set.

For any SNP $s$, let $P_s$ denote the set of distinct allele permutations at SNP locus $s$. Throughout we are indifferent to the order of each chromosome, with this in mind we can see that the total number of phases is bounded below by $\frac{1}{k!}\prod_s C(s)$.

## 2.2 Likelihood of a Phase

We formulate the haplotype reconstruction problem as identifying the most likely phase(s) given the read data $R$, all SNP loci $S$, as well as their genotypes, and sequencing error rates $\varepsilon$ . We assume the sequencing errors are independent of each other, that is for all $r \in R$ and all $s \in r$, that $\{r[s]\}$ are independently correct with probabilities $(1 - \varepsilon_{r,s})$ and incorrect with probabilities $\varepsilon_{r,s}$. Let $\varepsilon$ be a matrix containing all of these probabilities: $\{\varepsilon_{r,s}\}$. Given a vector set, $H$, corresponding to a phase, $R$, and $\varepsilon$; the likelihood of the phase is determined by:

$$\mathrm{P}[H \mid R, \varepsilon] = \frac{\mathrm{P}[R \mid H, \varepsilon\,]\,\mathrm{P}[H \mid \varepsilon]}{\mathrm{P}[R \mid \varepsilon]}. \tag{2.1}$$

As $\mathrm{P}[R \mid \varepsilon]$ depends only on $\varepsilon$ and the read set $R$, it is therefore the same across all vector sets. Hence, we define a *relative likelihood* measure (RL) as

$$\mathrm{RL}[H \mid R, \varepsilon] = \mathrm{P}[R \mid H, \varepsilon\,]\,\mathrm{P}[H \mid \varepsilon].$$

For $\mathrm{P}[H \mid \varepsilon]$, there are several ways this can be modeled depending on the situation. When we simulate polyploid data, we assume that $\mathrm{P}[H \mid \varepsilon]$ is equal for almost all vector sets, excluding those containing duplicate vectors. Let $M = \{m_1, m_2, ...\}$ be the set of the multiplicities in $H$; for example, if $H = \{0001, 0010, 1100, 0001, 0010\}$ then $M = \{2, 2, 1\}$. The probabilities $\mathrm{P}[H \mid \varepsilon]$ will differ multiplicatively by multinomial coefficients $\binom{k}{m_1, m_2, ...} = \frac{k!}{m_1! m_2! ...}$. Specifically:

$$\mathrm{P}[H \mid \varepsilon] = \frac{\binom{k}{m_1, m_2, \ldots}}{\prod_s C(s)}.$$

For the diploid case, there will never be duplicate vectors. To model $\mathrm{P}[h \in H \mid \varepsilon]$, we might assume that since mutations tend to occur together, adjacent SNP sites are more likely to be phased in parallel (00) or (11) than switched (01) or (10). Let $H = (h, h')$ and let $P(H)$ denote the number of adjacent SNPs that are parallel in $h$ and $S(H)$ the number of adjacent SNPs that are switched in $h$ (we must only consider $h$ as it determines $h'$). For example, if $H = ((00010111000), (11101000111))$, then $P(H) = 6$ and $S(H) = 4$. For some $p > .5$ (denoted as *parallel bias*) and $q = 1 - p$, we model this vector set probability as

$$\mathrm{P}[H \mid \varepsilon] = p^{P(H)} q^{S(H)}.$$

Finally, we consider $\mathrm{P}[R \mid H, \varepsilon]$. Because each read is independent, we can factor $\mathrm{P}[R \mid H, \varepsilon]$ over the probability $\mathrm{P}[r \mid H, \varepsilon]$ for each read $r \in R$.

$$\mathrm{P}[R \mid H, \varepsilon] = \prod_{r \in R} \mathrm{P}[r \mid H, \varepsilon]$$

To compute $\mathrm{P}[r \mid H, \varepsilon]$, for a given $r \in R$ and $h \in H$, let $A(r,h), D(r,h)$ denote the positions of SNP loci where $r$ and $h$ agree and disagree respectively. For example, if $r = (-,-,1,0,1,-,-,1,0)$ and $h = (1,0,0,1,1,0,0,1,0)$, then $A(r,h) = (5,8,9)$ and $D(r,h) = (3,4)$. We may now compute the desired probability, that is:

$$\mathrm{P}[r \mid H, \varepsilon] = \frac{1}{k} \sum_{h \in H} \left( \prod_{s \in A(r,h)} (1 - \varepsilon_{r,s}) \prod_{s \in D(r,h)} \varepsilon_{r,s} \right).$$

## Modification for Unknown Genotypes, Multi-allelic SNPs, and Partially Known Haplotypes

To generalize this model to incorporate either unknown genotypes, multi-allelic SNPs, or partially known haplotypes, we only must change the priors $\mathrm{P}[H \mid \varepsilon]$. In the case of unknown genotypes or multi-allelic SNPs, there are many more possible haplotypes, but we can still model $\mathrm{P}[H \mid \varepsilon]$ to be uniform aside from differing multiplicatively by $\binom{k}{m_1, m_2, \ldots}$, where the $m_i$ correspond to the multiplicities of vectors within the vector set.

For the case of partially known haplotypes $\tilde{H}$, suppose the haplotypes are known only at the set of locations $\tilde{S} \subseteq S$. One approach for assigning prior probabilities to vector sets (or haplotypes) is to assign a probability of 0 to any $H$ which is incompatible with $\tilde{H}$. Perhaps more interestingly, in the diploid bi-allelic case, one could assign a penalty for each switch error which must occur between the proposed solution $H$ and $\tilde{H}$. One may do so, for example, by assigning probabilities $\mathrm{P}[H \mid \varepsilon]$ proportional to $\gamma^{-M}$ where $\gamma \in (0, 1)$ and $M$ is the minimum number of switch errors between $H, \tilde{H}$.

# Chapter 3

# Relative Likelihood Model: HapTree-X

HapTree-X is a single individual haplotype assembly algorithm for diploid genomes and transcriptomes. Genomic read data differs from transcriptomic read data in several important and useful ways; we discuss these differences and provide notation describing them in order to ultimately define a relative likelihood model which successfully handles the transcriptomic and genomic cases concurrently.

## 3.1 Definitions and Notation

The goal of phasing is to recover the unknown haplotypes (haploid genotypes), $H = (H_0, H_1)$, which contain the sequence of variant alleles inherited from each parent of the individual. As homozygous SNPs are irrelevant for phasing, we restrict ourselves to heterozygous SNPs (from now on referred to simply as a 'SNP') and we denote the set of these SNPs as $S$. We assume these SNPs to be biallelic, and because of these restrictions, $H_0$ and $H_1$ are complements. Let $H[s] = (H_0[s], H_1[s])$ denote the alleles present at $s$, for $s \in S$.

In genomic read data (DNA-seq), all $r \in R$ (the set of reads) are equally likely to be sampled from the maternal or paternal chromosomes. In transcriptomic read data (RNA-seq) however, this may not always be the case. There are several ways to discuss the differing rates of maternal and paternal expression; we define some here. Furthermore, we can leverage this bias to perform phasing in certain cases. The blocks which can be phased in this case are larger that those which can be phased without using RNA-seq data. We generalize the Read Graph as defined in section 2.1 below to include the additional SNPs which may be phased using HapTree-X.

Some genes have multiple isoforms (alternative splicings) complicating phasing based on differential expression. In this chapter we phase using differential expression only for genes without multiple isoforms, or for subsets of SNPs $S$ within a gene with multiple isoforms such that the sets of isoforms covering each SNP $s \in S$ are identical. For an investigation

into the more general case of multiple isoforms, see Chapter 6.

## Differential Haplotypic Expression (DHE)

In this thesis, we define the differential haplotypic expression (DHE) to represent the underlying *expression bias* between the maternal and paternal chromosomes of a particular gene. Throughout, we will refer to the probability of sampling from the higher frequency haplotype of a gene as $\beta$. We assume two genes $g, g'$ have independent expression biases $\beta, \beta'$. A toy example gene featuring differential haplotypic expression is in figure 3.1 below.

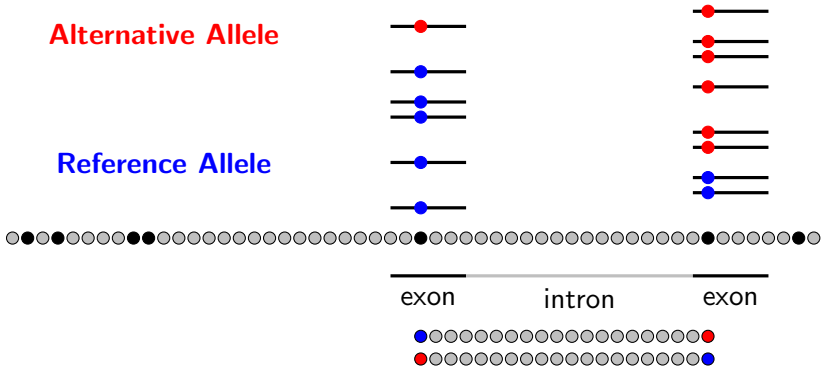


**Figure 3.1:** Differential Haplotypic Expression Example

## Differential Allele-Specific Expression (DASE)

Differential allele-specific expression (DASE) we define as the *observed* bias in the alleles at a particular SNP locus present in $R$.

| Allele \ SNP | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| **0** | 12 | 15 | 79 | 97 | 11 |
| **1** | 92 | 85 | 7 | 4 | 84 |

**DASE: Observed Allele Counts**

**Guess of True Haplotype**

**1** 2 3 4 5
0 0 1 1 0
1 1 0 0 1

**Guess of DHE**
$\beta = .9$

## Concordant Expression

We define *concordant expression* as when the DASE of a SNP agrees with the DHE of the gene to which the SNP belongs; that is when the majority allele (allele occurring with higher

frequency) occurring within the reads at a particular SNP locus is in agreement with the expected majority allele as determined by the DHE. In the case of concordant expression, the true DHE signal of the gene is being observed within the data. When this is not the case, we say we have *discordant expression*. For examples of each, please see figure 3.2 below.

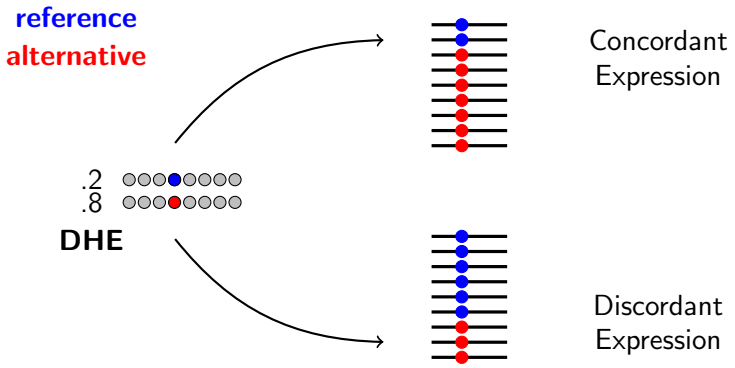


**Figure 3.2:** Concordant Expression and Discordant Expression Example

## Blocks to be Phased: Joint Read Graph

### DASE-based Phasing Blocks

To phase using differential expression (DASE-based phasing), we assume the existence of some gene model $G$ that specifies the genes (and their exons) within the genome. For each $g \in G$, we assume that the haplotypes $(H_0, H_1)$ restricted to $g$ are expressed at rates $\beta_0, \beta_1$ respectively due to DHE. The phasing blocks correspond to the SNPs in genes $g \in G$, though we will see that some SNPs are not phased due to insufficient probability of concordant expression. Two distinct genes $g, g'$ may not be DASE-phased due to lack of correlation between their expression biases $\beta, \beta'$. In the remainder of this thesis, when DASE-phasing a particular gene, by $H$ we mean the gene haplotype, that is $H$ restricted to the SNPs within $g$.

### Contig-based Phasing Blocks

To perform phasing using the sequence contiguity within reads (contig-based phasing), upon the set of SNP loci $S$ and read set $R$, we define a *Read Graph* such that there is a vertex for each SNP locus $s \in S$ and an edge between any two vertices $s, s'$ if there exists some read $r$ containing both $s$ and $s'$. These connected components correspond to the haplotype blocks to be phased.

### DASE-based and Contig-based Phasing Blocks

The blocks which are able to be phased by HapTree-X integrating both contig and DASE-based phasing are defined as the connected components of a *Joint Read Graph*. In the Joint Read Graph, each vertex corresponds to a SNP phased by either method, and there is an edge between any two vertices (SNPs) $s, s'$ if there exists some block that was phased by either method containing both $s, s'$.

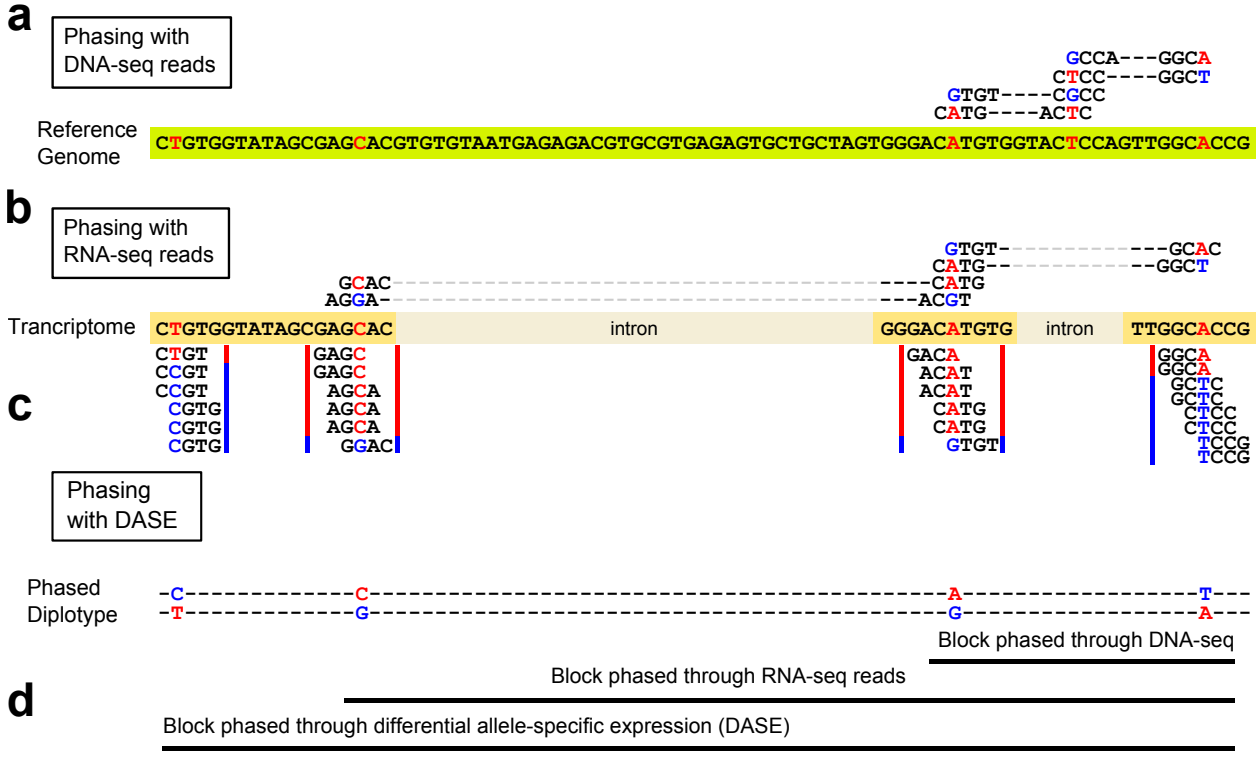


**Figure 3.3:** A toy example demonstrating the haplotype phasing capabilities of and differences between single-individual haplotype reconstruction methods using genome sequencing (DNA-seq) reads (a), transcriptome sequencing (RNA-seq) reads (b), and differential allele-specific expression (DASE) information that can be inferred from RNA-seq data (c).

In figure 3.3 above, green and orange blocks respectively represent reference genome and the transcriptome sequence, which contains only the exons in a gene separated by introns. Positions marked in red denote heterozygous-SNP loci. Paired-end sequencing reads are of length 2x4bp and have 3-4 bp insert lengths; reference alleles overlapping SNP loci are marked with red and alternative alleles are marked with blue. (a) Phasing using DNA-seq reads can be performed by looking at reads that overlap multiple heterozygous-SNP loci and observing the alleles that are connected through reads. Phasing distance is limited by maximum fragment length (12bp in the example). Multiple SNP loci can be chained together for phasing, but the probability of a switch error increases with the length of the chain. (b)

Though limited to only the SNPs within the transcriptome, RNA-seq reads have longer distance phasing capability than DNA-seq reads due to long introns in the genome that are spliced-out in the sequenced transcript fragments. RNA-seq reads also provide higher accuracy phasing of SNPs within the transcriptome compared to DNA-seq, since DNA-seq phasing needs to chain through intron SNPs to connect the exons. (c) Differential allele-specific expression (DASE) at transcriptomic SNP loci is available within RNA-seq datasets in the form of allele-specific coverage ratios. For genes that display differential haplotypic expression (DHE), the majority of alleles can be phased together to obtain a single haplotype block for the entire gene. Depending on the DHE and depth-coverage, DASE-based phasing can perform accurate haplotype reconstruction, independent of gene/exon lengths, without requiring paired-end or long reads. (d) Phasing capabilities of DNA-seq, RNA-seq and DASE-based phasing methods are demonstrated on the given toy example. The genome sequencing based approach is only able to provide haplotype blocks for the exons close together. The RNA-seq read based approach is able to reconstruct a longer haplotype block, phasing through the introns as well, but failing to phase far apart SNPs within the first exon. Whereas DASE-based phasing is able to reconstruct the complete gene haplotype by leveraging differential expression at SNP loci.

## 3.2 Likelihood of a Phase

We formulate the haplotype reconstruction problem as identifying the most likely phase(s) of set of SNPs $S$, given the read data $R$, and sequencing error rates $\varepsilon$. Furthermore, suppose we knew for each read $r$, the likelihood that $r$ was sampled from $H_i$ (denote this as $\beta_i^r$); we represent these probabilities as a matrix $\mathcal{B}$. While $\mathcal{B}$ is not given to us, we may estimate $\mathcal{B}$ from $R$ (see section 3.3). We derive a likelihood equation for $H$, conditional on $R, \mathcal{B}$ and $\varepsilon$.

Given a haplotype $H$, reads $R$, error rates $\varepsilon$, and $\mathcal{B}$, the probability of $H$ being the true phase is

$$\mathrm{P}[H \mid R, \mathcal{B}, \varepsilon\,] = \frac{\mathrm{P}[R \mid H, \mathcal{B}, \varepsilon\,]\,\mathrm{P}[H \mid \mathcal{B}, \varepsilon]}{\mathrm{P}[R \mid \mathcal{B}, \varepsilon\,]}. \tag{3.1}$$

Since $\mathrm{P}[R \mid \mathcal{B}, \varepsilon]$ does not depend on $H$, we may define a *relative likelihood* measure, RL. Note that $\mathrm{P}[H|\mathcal{B}, \varepsilon] = \mathrm{P}[H]$ as the priors on the haplotypes are independent of the errors in $R$, and of $\mathcal{B}$.

$$\mathrm{RL}[H \mid R, \mathcal{B}, \varepsilon\,] = \mathrm{P}[R \mid H, \mathcal{B}, \varepsilon\,]\,\mathrm{P}[H \mid \mathcal{B}, \varepsilon]. \tag{3.2}$$

For the prior $\mathrm{P}[H \mid \mathcal{B}, \varepsilon]$, we assume a potential *parallel bias*, $\rho \geq .5$, (the prior probability of adjacent SNPs being phased in parallel as opposed to switched) which results in

a distribution on $H$ such that adjacent SNPs are independently believed to be phased in parallel $\binom{00}{11}$ with probability $\rho$ and switched $\binom{01}{10}$ with probability $1-\rho$. When $\rho = .5$ we have the uniform distribution on $H$. The general prior distribution on $H$ in terms of $\rho$ is (here we mean either $H_0$ or $H_1$ as they are conjugate).

$$\mathrm{P}[H] = \rho^{P(H)}(1-\rho)^{S(H)} \tag{3.3}$$

where $P(H)$ and $S(H)$ denote the number of adjacent SNPs that are parallel and switched in $H$, respectively.

Given the above model, as each $r \in R$ independent, we may expand $\mathrm{P}[R \mid H, \mathcal{B}, \varepsilon\,]$ as a product:

$$\mathrm{P}[R \mid H, \mathcal{B}, \varepsilon] = \prod_{r \in R} \mathrm{P}[r \mid H, \mathcal{B}, \varepsilon] \tag{3.4}$$

In the setting of RNA-seq, reads are not sampled uniformly across homologous chromosomes, but rather according to the DHE (*expression bias*) of the gene from which they are transcribed. We see in (3.5) how this asymmetry allows us to incorporate reads which contain only one SNP. Let $A(r, H_i), D(r, H_i)$ denote the SNP loci where $r$ and $H_i$ agree and disagree respectively, then

$$\mathrm{P}[r \mid H, \mathcal{B}, \varepsilon] = \sum_{i \in [0,1]} \beta_i^r \left( \prod_{s \in A(r,H_i)} (1 - \varepsilon_{r,s}) \prod_{s \in D(r,H_i)} \varepsilon_{r,s} \right). \tag{3.5}$$

When there is uniform expression $\beta_0^r = \beta_1^r$ (no bias) and if $|r| = 1$, then $\mathrm{P}[r \mid H, \mathcal{B}, e]$ is constant across all $H$. This is not the case when the expression bias is present however, and therefore reads covering only one SNP affect the likelihood of $H$.

If we knew the matrix $\mathcal{B}$, we could apply HapTree [7] to search for $H$ of maximal likelihood; the matrix $\mathcal{B}$, however, is unknown. Suppose instead we are given some probability distribution for the entries of $\mathcal{B}$, to compute $\mathrm{P}[r \mid H, \mathcal{B}, \varepsilon]$, it is enough to know the expected value of each entry because of the linearity (over $i$) of $\mathrm{P}[r \mid H, \mathcal{B}, \varepsilon]$. To this aim, we provide methods for determining a maximum likelihood $\mathcal{B}$. To approximate distributions for the entries of $\mathcal{B}$, we assume for each gene there is uniform expression with some probability $p$, and differential expression with probability $1-p$; in the latter case, the differential expression is assumed to be that of maximal likelihood. By varying $p$, we can vary the relative weights associated to DASE-based phasing and contig-based phasing.

Furthermore, we develop methods for determining for which reads $r$ we are sufficiently confident there this is in fact non-uniform expression, that is $\beta_0^r \neq \beta_1^r$. Moreover, we determine for which SNPs $s \in S$ (contained only by reads of size one), we have sufficient coverage and expression bias to determine (with high accuracy) the phase $H[s]$.

## 3.3 Maximum Likelihood Estimate of Differential Haplotypic Expression

For a fixed gene $g$, containing SNPs $S_g$, the corresponding reads $R_g$ have expression biases $\beta_0^r, \beta_1^r$ which are constant across $r \in R_g$. Let $\beta = \beta_0^r$ refer to this common expression; we wish to determine the maximum likelihood underlying expression bias $\beta$ of $g$ responsible for producing $R_g$. To do so, we formulate a Hidden Markov Model (HMM) and use the forward algorithm to compute relative likelihoods of $R$ given $\beta, \varepsilon$.

To achieve the conditional independence required in a HMM, we define $R'_g$, a modification of $R_g$, containing only reads of size one, so that $R'_{g,s}$ (the reads $r \in R'_g$ which cover $s$) are independent from $R'_{g,s'}$ ($\forall s \neq s' \in S_g$). We restrict each $r \in R_g$ to a uniformly random SNP $s$, and include this restricted read of size one ($r|_s$) in $R'_g$ (we note that if $|r| = 1$, then $r = r|_s$, by definition.) Therefore, $R'_{g,s}$ and $R'_{g,s'}$ are independent as all $r \in R'_g$ are of size one.

Our goal is the determine the maximum likelihood $\beta$, given $R'_g$. We assume a uniform prior on $\beta$, and therefore $\mathrm{P}[\beta \mid R'_g, \varepsilon]$ is proportional to $\mathrm{P}[R'_g \mid \beta, \varepsilon]$ (immediate from Bayes theorem). We may theoretically compute $\mathrm{P}[R'_g \mid \beta, \varepsilon]$ by conditioning on $H$ (which is independent from $\beta, \varepsilon$)

$$\mathrm{P}[R'_g \mid \beta, \varepsilon] = \sum_H \mathrm{P}[R'_g \mid H, \beta, \varepsilon] \, \mathrm{P}[H]$$

and expand $\mathrm{P}[R'_g \mid H, \beta, \varepsilon]$ as a product over $r \in R'_g$ as in (3.4) and (3.5). This method, however, requires enumerating all $H$; since $|H| = 2^{|S_g|}$ we seek different approach. Indeed, we translate this process into the framework of a Hidden Markov Model, apply the forward algorithm to compute $f(\beta) := \mathrm{P}[R'_g \mid \beta, \varepsilon]$ exactly for any $\beta$, and since $f$ has a unique local maxima for $\beta \in [.5, 1]$, we can apply Newton-Rhapson method to determine $\beta$ of maximum likelihood.

### HMM Model Translation

To set this problem in the framework of a hidden Markov model, we let the haplotypes $H$ correspond to the hidden states, $R'_g$ to the observations, and let the time evolution be the ordering of the SNPs $S_g$. The observation at time $s$ in this context is $R'_{g,s}$, the reads covering SNP $s$. The emission distributions are as follows:

$$\mathrm{P}[R'_{g,s} \mid H[s], \beta, \varepsilon] = \prod_{r \in R'_{g,s}} \mathrm{P}[r \mid H[s], \beta, \varepsilon]$$

$$\mathrm{P}[r \mid H[s], \beta, \varepsilon] = \begin{cases} \beta_0(1 - \varepsilon_{r,s}) + (1 - \beta_0)\varepsilon_{r,s} & \text{if } r[s] = H_0[s] \\ \beta_1(1 - \varepsilon_{r,s}) + (1 - \beta_1)\varepsilon_{r,s} & \text{if } r[s] = H_1[s] \end{cases} \tag{3.6}$$

where $H[s]$ is $H$ restricted to $s$.

To determine the hidden state transition probabilities, recall our prior on $H$ in (3.3). We may equivalently model this distribution $H$ as a Markov chain, with transition probabilities:

$$\mathrm{P}[H[s_{i+1}] \,|\, H[s_i]] = \begin{cases} \rho & \text{if } H_0[s_i] = H_0[s_{i+1}] \\ 1-\rho & \text{if } H_0[s_i] \neq H_0[s_{i+1}] \end{cases}$$

These emission probabilities and hidden state transition probabilities are all that are needed to apply the forward algorithm and determine the $\beta$ of maximum likelihood. The diagram below in figure 3.4 shows the underlying graphical structure of this hidden Markov model.

$$\gamma_s(\beta) : \mathrm{P}[R'_{g,s} \,|\, H[s], \beta, \varepsilon] \qquad\qquad \alpha_{s_i,s_{i+1}} : \mathrm{P}[H[s_{i+1}] \,|\, H[s_i]]$$

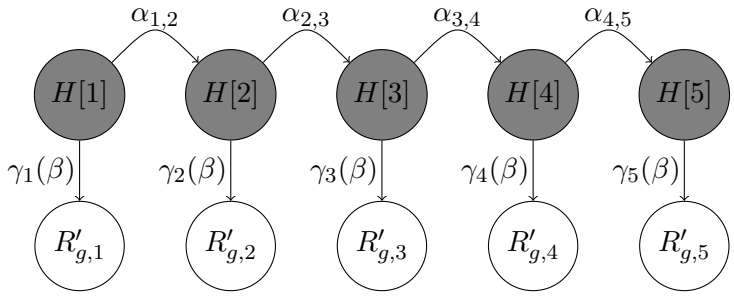


**Figure 3.4:** Hidden Markov Model for Estimating Differential Haplotypic Expression

## 3.4 Likelihood of Concordant Expression

### A Solution of Maximum Likelihood

In this section we prove that the intuitively correct solution (under mild conditions CND1, CND2, and CND3) is that of maximal likelihood. In doing so, we see the role played by concordant expression, and motivate its use as a probabilistic measure for determining which SNPs we believe we may phase with high accuracy.

We derive $H^+$, a haplotype solution of a gene $g$, of maximum likelihood given $R'_g$, $\beta$ and $\varepsilon$ and conditions CND1, CND2, and CND3. These conditions are:

**Conditions:**

- CND1: Error rates are constant (say $\epsilon$)
- CND2: Error rates less than a half ($\epsilon < .5$)
- CND3: Uniform prior distribution ($\rho = .5$)

Let $C_s^v$ denote the number of reads $r \in R'_{g,s}$ such that $r[s] = v$ where $v \in \{0, 1\}$. Provided error rates are constant (CND1) (say $\epsilon$) and $\epsilon < .5$ (CND2), and assuming a uniform prior distribution ($\rho = .5$) (CND3), we can show a solution of maximum likelihood is $H^+ = (H_0^+, H_1^+)$, where $H_0^+[s] = v$ such that $C_s^v \geq C_s^{1-v}$. In words, $H_0^+$ and $H_1^+$ contain the alleles that are expressed the majority and minority of the time (respectively) at each SNP locus; given sufficient expression bias and coverage, intuitively, $H^+$ ought to correctly recover the true haplotypes. It is easy to show that CND1 and CND3 can be removed if one is willing to specify a minimum coverage; we do not show this here. Intuitively, CND2 must not be removed.

To prove $H^+$ is of maximal likelihood, we introduce the terms *concordant expression* and *discordant expression*. We say $R$ and $H$ have *concordant expression* at $s$ if $C_s^{H_0[s]} > C_s^{H_1[s]}$, *discordant expression* if $C_s^{H_0[s]} < C_s^{H_1[s]}$, and *equal expression* otherwise. In words, since we assume $\beta_0 > \beta_1$, we *expect* to see the allele $H_0[s]$ expressed more than the allele $H_1[s]$ in $R_{g,s}$ (concordant expression.)

We may now equivalently define $H^+$ as a solution which assumes concordant or equal expression at every SNP $s$. Because we assume uniform priors, $\mathrm{P}[H \mid R'_g, \beta, \epsilon]$ is proportional $\mathrm{P}[R'_g \mid H, \beta, \epsilon]$ (see (3.1)), and since each read is of size one, we can factor across $S_g$ in the following way:

$$\mathrm{P}[R \mid H, \beta, \epsilon] = \prod_{s \in S_g} \mathrm{P}[R_{g,s} \mid H[s], \beta, \epsilon]$$

Therefore, to show $H^+$ is of maximal likelihood, it only remains to show that concordant expression is at least as likely as discordant expression, as intuition suggests. Let $\gamma_i = \beta_i(1 - \epsilon) + (1 - \beta_i)\epsilon$, then as in (3.6) we may deduce

$$\mathrm{P}[R_{g,s} \mid H[s], \beta, \epsilon] = \prod_{i \in \{0,1\}} \gamma_i^{C_s^{H_i[s]}}$$

Let $H^- = (H_1^+, H_0^+)$, the opposite of $H^+$. We can now compare the likelihood of concordant (or equal) expression at $s$ ($H^+[s]$) with that of discordant (or equal) expression at $s$ ($H^-[s]$.) For ease of notation, let $v_i = H_i^+[s]$ and $w_i = H_i^-[s]$.

$$\frac{\mathrm{P}[R_{g,s} \mid H^+[s], \beta, \epsilon]}{\mathrm{P}[R_{g,s} \mid H^-[s], \beta, \epsilon]} = \frac{\prod_{i \in \{0,1\}} \gamma_i^{C_s^{v_i}}}{\prod_{i \in \{0,1\}} \gamma_i^{C_s^{w_i}}} = \frac{\gamma_0^{C_s^{v_0} - C_s^{w_0}}}{\gamma_1^{C_s^{w_1} - C_s^{v_1}}} = \left(\frac{\gamma_0}{\gamma_1}\right)^{C_s^{v_0} - C_s^{v_1}} \geq 1 \tag{3.7}$$

The rightmost equality results from the fact that $H_i^+ = H_{1-i}^-$, and hence $v_i = w_{1-i}$. Since $\epsilon < .5$, we have $\gamma_0 > \gamma_1$; $C_s^{v_0} - C_s^{v_1} \geq 0$ by the definition of $H^+$, which proves the inequality.

## Computing Likelihood of Concordant Expression

We just showed that under mild conditions, the solution of maximal likelihood is, intuitively, that which has concordant expression at each SNP locus $s$. Therefore, to determine which SNPs we believe we can phase with high accuracy, we measure the probability of concordant expression at that SNP, and only phase when that probability is sufficiently high.

The probability of concordant expression can be immediately derived from (3.7). We assume a uniform error rate of $\epsilon$ for ease of notation, though is not required. Let $\mathrm{CE}(R_{g,s}, H[s])$ denote the event of concordant expression at $s$, then

$$\mathrm{P}[\mathrm{CE}(R_{g,s}, H[s]) \mid \beta, \epsilon] = \frac{\mathrm{P}[R_{g,s} \mid H^+[s], \beta, \epsilon]}{\mathrm{P}[R_{g,s} \mid H^+[s], \beta, \epsilon] + \mathrm{P}[R_{g,s} \mid H^-[s], \beta, \epsilon]} \tag{3.8}$$

$$= \frac{1}{1 + \left(\frac{\gamma_1}{\gamma_0}\right)^{|C_s^0 - C_s^1|}} \tag{3.9}$$

Furthermore, given $N$ reads, an expression bias $\beta$, and a constant error rate $\epsilon$, we compute likelihood of concordant expression using the standard binomial distribution $B(N, \gamma_0)$ by equating 'successes' in the binomial model to observations of the majority allele, expressed with bias $\gamma_0$ (recall $\gamma_i$ takes errors into account):

$$\mathrm{P}[\mathrm{CE} \mid N, \beta, \epsilon] = \sum_{i=\lceil \frac{N+1}{2} \rceil}^{N} \binom{N}{i} \gamma_0^i \gamma_1^{N-i} \;\geq\; 1 - e^{-N \frac{1}{2\gamma_0} (\gamma_0 - \frac{1}{2})^2} \tag{3.10}$$

To obtain the bound on the right hand side, apply the Chernoff bound $\mathrm{P}[X < (1-\lambda)\mu)] \leq e^{-\frac{\lambda^2 \mu}{2}}$ where $X$ corresponds to the number of 'successes' and $\mu = \mathrm{E}[X] = N\beta$. This bound shows that the probability of concordant expression increases exponentially with the coverage $(N)$.

We remark for large $N$, the Binomial Distribution $\mathrm{B}(n, \beta)$ converges to the normal distribution $\mathcal{N}(N\beta, N\beta(1-\beta))$, and therefore this probability can always be easily computed. See Figures 5.10 and 5.11 for a sense of these likelihoods.

## Likelihood of Non-Biased Expression

Now that we have a method for determining the likelihood of concordant expression, we can require any SNP loci to have a sufficiently high probability of concordant expression in order for HapTree-X to attempt to phase that SNP. The likelihood of concordant expression is dependent on $\beta$ however, which we may only estimate. We therefore require that for any gene $g$ (and SNPs within it) to be phased, the differential allele-specific expression within the gene must be sufficiently unlikely to have been generated by uniform DHE ($\beta = .5$) (because

in this case, we can not use DASE-based methods to phase). We compute an upper bound on this probability using a two sided binomial test applied to total allele counts $m, M$, where

$$m = \sum_{s \in g} \min(C_s^0, C_s^1)$$

$$M = \sum_{s \in g} \max(C_s^0, C_s^1).$$

The likelihood of at least $M$ heads and at most $m$ tails is computed below. Let $N = m + M$, then the upper bound based on the two-sided binomial test is

$$\sum_{i=0}^{m} \binom{N}{i} \frac{1}{2}^N + \sum_{i=M}^{N} \binom{N}{i} \frac{1}{2}^N.$$

As mentioned above, the Binomial Distribution $\mathrm{B}(n, \frac{1}{2})$ converges to the normal distribution $\mathcal{N}(\frac{N}{2}, \frac{N}{4})$, and therefore we may efficiently compute these likelihoods.

# Chapter 4

# Maximization of Relative Likelihood Score

## 4.1 Background

The goal of our haplotype reconstruction problem is to find the (sets of) haplotypes which maximize the product $\mathrm{P}[R \,|\, H, \varepsilon\,]\,\mathrm{P}[H \,|\, \varepsilon]$, equivalently $\mathrm{RL}[H|\, R, \varepsilon]$. The number of possible haplotypes is $O(2^{kn})$ (if we assume SNPs are bi-allelic, or $O(4^{kn})$ in the most general case) and therefore, in general, checking all possible solutions is intractable. Our solution is based on finding high-likelihood phases for the first $m$+1 SNPs, conditioned on a collection of high likelihood phases for the first $m$ SNPs.

### Enumerative Approach

For any fixed block of size $n$, one can enumerate all possible haplotype solutions and for each possible solution compute $\mathrm{RL}[H|\, R, \varepsilon]$. Suppose the set of reads $R_B$ cover the block $B$, and each $r \in R_B$ covers $\mathrm{l}(r)$ SNPs. For any fixed vector set $H$, the time to compute $\mathrm{RL}[H|\, R_B, \varepsilon]$ can be written as

$$k \sum_{r \in R_B} \mathrm{l}(r).$$

Summing over all blocks $B$, we see the total time to compute $\mathrm{RL}[H|\, R, \varepsilon]$ for a fixed vector set is proportional the total coverage of $R$:

$$k \sum_{B} \sum_{r \in R_B} \mathrm{l}(r) = k \sum_{R} \mathrm{l}(r).$$

The above holds as $R = \bigsqcup_B R_B$.

Unfortunately, to compute the probabilities for all possible haplotypes, exponentially many solutions must be enumerated. For sufficiently small $k$ and $n$, this is tractable, but even in the diploid case $k = 2$, block lengths can be in the hundreds, making the enumerative approach not fully applicable, motivating the algorithm HapTree.

## 4.2 Notation for HapTree and HapTree-X

For the description of our method, we assume known-genotypes and that SNPs are bi-allelic. After describing our method we also describe the changes needed to our original approach to accommodate multi-allelic and genotype-oblivious polyploid haplotype assembly.

### Semi-Reads and Sub-Reads

To properly describe our method we must first define the *semi-reads* of a SNP locus $s$ and the *sub-reads* of a subset $S' \subset S$.

#### Semi-reads

To form the set of *semi-reads* of $s$, denoted $SR(s)$, include each read $r \in R$ that contains both $s$ and some $s' < s$ ($s'$ is upstream of $s$) and ignore all information from $r$ on SNPs $s'' > s$ ($s''$ is downstream of $s$). Suppose the set of reads is:

{1:1, 2:1, 3:1, 4:1} {3:1, 4:1, 6:0} {4:0, 5:1, 6:1}
{4:0, 5:1, 6:1, 7:0} {5:0, 6:0, 7:1} {5:1, 6:1, 7:0}

The corresponding semi-reads for each SNP locus would be:

1 → None  2 → {1:1, 2:1}  3 → {1:1, 2:1, 3:1}
4 → {1:1, 2:1, 3:1, 4:1}  {3:1, 4:1}
5 → {4:0, 5:1}  {4:0, 5:1}
6 → {3:1, 4:1, 6:0}  {4:0, 5:1, 6:1}  {4:0, 5:1, 6:1}  {5:0, 6:0}  {5:1, 6:1}
7 → {4:0, 5:1, 6:1, 7:0}  {5:0, 6:0, 7:1}  {5:1, 6:1, 7:0}

#### Sub-reads

The *sub-reads* of $S' \subset S$, denoted $R(S')$, are obtained by, for each $r \in R$, removing all keys $s \in S \setminus S'$ to form $r'$, and then adding $r'$ to $R(S')$ if the length of $r'$ is at least 2. Alternatively, $R(S')$ corresponds to the set of reads relevant to the problem of only phasing $S'$. Continuing with the example above, if $S' = \{1, 2, 3, 4, 5\}$, then

$$R(S') = \{\{1{:}1,\ 2{:}1,\ 3{:}1,\ 4{:}1\},\ \{3{:}1,\ 4{:}1\},\ \{4{:}0,\ 5{:}1\},\ \{4{:}0,\ 5{:}1\}\}.$$

## 4.3 HapTree and HapTree-X Maximization Algorithm

Our main approach to solving the single individual polyploid haplotype assembly problem is by finding highly probable solutions on $m$ SNPs and extending those to highly probable

solutions on $m$+1 SNPs. Our algorithm has two fundamental parts: branching and pruning. For each connected component of the *ReadGraph*, $G(S, R)$, we inductively generate a collection of high likelihood phases on the first $m$ SNPs. For each of these phases, we branch them to phases on $m$+1 SNPs by considering all possible orderings of alleles for position $m$+1 and including branches for those which occur with probability above a certain threshold. After doing so, we prune the tree of phases by removing all leaves that occur with probability sufficiently less than the most probable leaf. We discuss both parts in more detail below. We note that although a dynamic programming algorithm can be directly applied to infer the best solutions under HapTree's likelihood model, we instead developed HapTree, which is substantially faster than exact dynamic programming but with nearly identical empirical performance.

## Extension

We first describe how to extend an existing a haplotype assembly $H$ on $m \geq 0$ SNPs onto the $m$+$1^{th}$ SNP $s$. Recall the set of permutations of $s$ is denoted $P_s$ and one particular permutation as $o \in P_s$. An extension $H'$ of $H$ onto SNP locus $s$ can be defined by appending some permutation $o \in P_s$ of alleles to $H$; $H' = H + o$. Note that it is possible for two distinct permutations to result in the same $H'$: $H + o = H + o'$. In these cases we do not include duplicates, as they are equivalent. Observe that if $H$ is empty, all allele permutations are the same as vector sets; we therefore include only one. For any $H'$, we can compute the probability of it being the correct haplotype (for the first $m$+1 SNPs) conditioning on $H$ being correct (for the first $m$ SNPs), as well as the semi-read data $SR(s)$ and error rate $\varepsilon$. We express this below:

$$\mathrm{P}[H' \mid H,\, SR(s),\, \varepsilon] = \frac{\mathrm{P}[SR(s) \mid H',\, H,\, \varepsilon]\, \mathrm{P}[H' \mid H,\, \varepsilon]}{\mathrm{P}[SR(s) \mid H,\, \varepsilon]} \tag{4.1}$$

This computation is similar to those done above in equation (2.1). The EXTEND algorithm (Algorithm 1) is given below, which returns a list of all extensions $H'$ of $H$ that occur with probability above a certain threshold, $\rho$, given haplotype $H$.

**Input:** $H, \rho, s$
**Output:** $E'$
  $E = [\,]$
  **for** $o \in P_s$ **do**
    $H' = H + o$
    **if** $H' \notin E$ **then**
      **if** $\mathrm{P}[H' \mid H,\, SR(s),\, \varepsilon] > \rho$ **then**
        $E$ += $H'$
  **return** $E$

**Algorithm 1:** EXTEND($H, \rho, s$): Extending a haplotype $H$ at SNP $s$ to all $H'$ that occur with probability $\geq \rho \in [0, 1)$.

## Branching

Here we define *branching* a collection of haplotypes $\mathcal{H}$ with threshold $\rho$ to SNP $s$: BRANCH($\mathcal{H}, \rho, s$) (Algorithm 2) . We assume all $H \in \mathcal{H}$ phase the first $m \geq 0$ SNPs and that SNP $s$ is the $m{+}1^{th}$ SNP. The act of branching $\mathcal{H}$ returns $\mathcal{H}'$: a list of all extensions generated by EXTEND with threshold $\rho$ for all $H$ in $\mathcal{H}$. To initialize BRANCH we EXTEND the empty vector set to an arbitrary permutation of the alleles of the first SNP, as all permutations are equivalent as vector sets.

**Input:** $\mathcal{H}, \rho, s$
**Output:** $\mathcal{H}'$
  $\mathcal{H}' = [\,]$
  **for** $H \in \mathcal{H}$ **do**
    $E = \text{EXTEND}(H, \rho, s)$
    **for** $H' \in E$ **do**
      $\mathcal{H}$ += $H'$
  **return** $\mathcal{H}$

**Algorithm 2:** BRANCH($\mathcal{H}, \rho, s$): Branching haplotypes $\mathcal{H}$ at SNP $s$ with threshold $\rho \in [0, 1)$.

## Pruning

For a collection of haplotypes $\mathcal{H}$ of SNPs $S' \subset S$, we can compute the relative likelihood of each haplotype conditioned on the sub-reads $R(S')$ and error rate $\varepsilon$; we write this as $\text{RL}[\, H \mid R(S'), \varepsilon]$. The same computation as performed in equation 2.1 yields:

$$\text{P}[H \mid R(S'), \varepsilon] = \frac{\text{P}[R(S') \mid H, \varepsilon\,]\,\text{P}[H \mid \varepsilon]}{\text{P}[R(S') \mid \varepsilon]}.$$

Since $\text{P}[R(S') \mid \varepsilon]$ does not depend on $H$:

$$\text{RL}[\, H \mid R(S'), \varepsilon] = \text{P}[R(S') \mid H, \varepsilon\,]\,\text{P}[H \mid \varepsilon]. \tag{4.2}$$

The goal of PRUNE($\mathcal{H}, \kappa, S'$) (Algorithm 3) is to return a subset $\mathcal{H}' \subset \mathcal{H}$ containing only sufficiently probable haplotypes. It does so by computing the relative likelihood of the most probable $H \in \mathcal{H}$, that is $\omega = \max_{H \in \mathcal{H}} \text{RL}[H \mid R(S'), \varepsilon]$, and adding $H \in \mathcal{H}$ to $\mathcal{H}'$ if $\text{RL}[H \mid R(S'), \varepsilon] \geq \kappa\omega$, where $\kappa$ is between 0 and 1. We note that that one can compute $\text{RL}(H')$ from $\text{RL}(H)$ by only looking at the semi-reads $RS(s)$: we store the relative likelihood values for all $H \in \mathcal{H}$ and update them when branching to $\mathcal{H}'$; PRUNE is therefore no more costly than BRANCH.

**Input:** $\mathcal{H}, \kappa, S'$
**Output:** $\mathcal{H}'$
  $\mathcal{H}' = [\,]$
  $\omega = \max_{H \in \mathcal{H}} \mathrm{RL}[H \mid R(S'),\, \varepsilon]$
  **for** $H \in \mathcal{H}$ **do**
    **if** $\mathrm{RL}[H \mid R(S'),\, \varepsilon] \geq \kappa\omega$ **then**
      $\mathcal{H}'$ += $H$
  **return** $\mathcal{H}$

**Algorithm 3:** PRUNE($\mathcal{H}, \kappa, S'$): Pruning haplotypes $\mathcal{H}$ on $S'$ with factor $\kappa \in$ [0,1] .

## Main Algorithm

Here we give a high-level description of our overall haplotype assembly method HapTree($R, \hat{\rho}, \hat{\kappa}, S$) (Algorithm 4) using the EXTEND, BRANCH, and PRUNE algorithms. We generate high likelihood phases for the first $m$ SNPs, BRANCH those phases to include $s$ (the $m$+1$^{th}$ SNP), then PRUNE the resulting phases, and repeat for $m = m+1$. We begin with an arbitrary permutation of the first SNP, since all orderings result in the same vector set. For the final step, we PRUNE with $\kappa = 1$, and therefore return only the maximally probable phases that we have found; if this set is of size greater than one, we choose a phasing from within it randomly. More generally, below we take $\hat{\rho}$ and $\hat{\kappa}$ to be vectors, as $\rho$ and $\kappa$ may depend on $m$, the size of $\mathcal{H}$ or other user-specified variables.

**Input:** $R, \hat{\rho}, \hat{\kappa}, S$
**Output:** $\mathcal{H}$
  $\mathcal{H} = [\,]$
  $S' = \{\}$
  **for** $s \in [1, 2, ..., |S|]$ **do**
    $S'$ += $s$
    $\mathcal{H} = \mathrm{BRANCH}(\mathcal{H}, \hat{\rho}(s), s)$
    $\mathcal{H} = \mathrm{PRUNE}(\mathcal{H}, \hat{\kappa}(s), S')$
  **return** $\mathcal{H}$

**Algorithm 4:** HapTree($R, \hat{\rho}, \hat{\kappa}, S$): Assembling haplotype from reads $R$ with parameters $\hat{\rho}, \hat{\kappa}$.

# Further Remarks

## Dynamic Programming Approach

We present an exact dynamic programming algorithm which we believe is first described in [18]. In [18], however, the author minimizes MEC score, as opposed to maximizing relative likelihood, and assumes gapless reads.

For phasing a block $B$, a simple dynamic programing algorithm exists with runtime $O(C \times n \times 2^m)$ (in the diploid bi-allelic case), where $C$ is the sum over all reads (covering $B$) of the number of SNPs covered by each, $n$ is the total number of SNPs in $B$, and $m$ is the maximum *range* of any read covering $B$. The range of a read is the total number of SNPs between (and including) the first SNP it covers and the last. For gapless reads, the range is equal to the *length* of the read (the number of SNPs covered by the read). For reads with gaps, the range is strictly more than the number of SNPs covered.

The algorithm consists of computing, for each $0 \leq i \leq n - m - 1$, the maximum log-probability of a haplotype solution of length $i + m$ with a particular suffix $H_m$. Let $M_i(H_m)$ denote this maximum probability, and let $R_i$ denote the set of all reads originating at SNP $i$. The key insight into the algorithm is

$$M_i(H_m) = \max_{H'_m} \left( M_{i-1}(H'_m) + \log(\mathrm{P}[R_i \mid H'_m\,,\, \varepsilon)] \right)$$

where $H'_m$ is any set of haplotypes of length $m$ whose length $m-1$ suffix is the length $m-1$ prefix of $H_m$. In the diploid bi-allelic case, there are two such $H'_m$, corresponding to the possible phases of a SNP $\binom{0}{1}$ and $\binom{1}{0}$. The algorithm is initialized with

$$M_0(H_m) = \log(\mathrm{P}[R_i \mid H_m\,,\, \varepsilon)].$$

We can trace back through these values to determine all haplotypes of maximal likelihood.

## Read Graph Restructuring Speed-Ups

A simple way to speed up HapTree (and the exact dynamic programming algorithm) can be seen by looking more carefully at the connectivity of the read graph $G$. A graph $G$ is *biconnected* if removing any vertex does not disconnect the graph. Any graph can be decomposed into biconnected components, and any pair of these components either share a vertex or are connected by a single edge. Moreover, decomposing a graph into its biconnected subgraphs can be done in linear time [19]. We remark that any solution of maximal likelihood can be found by finding restricting to the biconnected components of $G$ and finding maximal likelihood solutions restricted to those components, and trivially gluing them together (when a pair is connected by an edge, make the greedy choice).

There is a more complicated, and unfortunately NP-Hard [15], method which would in theory speed up the exact dynamic programming algorithm as well. For some connected component of $G$, say $B$, let $m$ denote the maximum *range* of any read covering $B$; the range

of a read is the total number of SNPs between (and including) the first SNP in a read and the last. In terms of the read graph, assign to each $v \in V$ an integer $f(v)$, where $f(v)$ is such that $v$ corresponds to the $f(v)^{th}$ SNP in $B$. Then $m = \max_E \left(|f(v) - f(v')| + 1\right)$ where $E$ is the set of edges $(v, v')$ of $G$. By permuting the ordering of the SNPs, $m$ can be minimized; this is known as the *graph bandwidth*. Any permutation reducing $m$ can be applied to the SNPs, resulting in faster runtime for exact dynamic programing algorithm above. Unfortunately, finding permutations which minimize $m$ and determining the bandwidth of a graph is known to be NP-Hard [15].

## Extending to Partially Known Haplotypes, Multi-allelic SNPs, and Unknown Genotypes

We remark that to generalize HapTree to handle partially known haplotypes, multi-allelic SNPs, and unknown genotypes, the only step in the algorithm that must be updated is Extension. In each case, the set $O_p$ of possible phases for the next SNP is modified. In the case of partially known haplotypes, we can restrict at the known SNP loci the size of $|O_p|$ to be one (to reflect the phase in the partially known haplotype); in this case, no solution that disagrees with the partially known haplotype will be returned. Alternatively, we may just modify the priors as discussed at the end of Chapter 2. For multi-allelic SNPs and unknown genotypes, $O_p$ is updated to reflect the larger set of possibilities, and the priors are updated accordingly.

# Chapter 5

# Simulated and Experimental Results

## 5.1 Summary

In this thesis, we introduced a maximum-likelihood formulation of the haplotype reconstruction problem in several instances and present a haplotype assembly algorithm, HapTree(-X), which concurrently performs SNP-pair phasing and full haplotype assembly based on a probabilistic framework. We demonstrate the performance of HapTree(-X) by looking at both common and innovate metrics for measuring phasing accuracy on both simulated and experimental data. We find that HapTree and HapTree-X outperform previous methods for solving the single individual haplotype assembly problem by increasing accuracy, speed, and in case of HapTree-X, number SNPs phased and length of phased blocks as well. We demonstrate these results in the following section.

### HapTree: Diploid

As a proof of concept, we test our method HapTree on real diploid sequencing data from NA12878 (1000 Genomes Project) and evaluate the quality of assembled haplotypes with respect to trio-based diplotype annotation as the ground truth. The results indicate that even for diploid genomes, HapTree improves the switch error accuracy within phased haplotype blocks as compared to existing haplotype assembly methods [5], while producing comparable MEC values.

### HapTree: Polyploid

Because polyploid data of an organism with an accurately phased genome to compare against is hard to come by, we evaluate the performance of HapTree on simulated polyploid sequencing read data modeled after Illumina and 454 sequencing technologies. For triploid and higher ploidy genomes, we demonstrate that HapTree substantially improves haplotype assembly accuracy and efficiency over the state-of-the-art HapCompass [3, 4] for varying read depth

coverage and length of haplotype block; moreover, HapTree is the first scalable polyplotyping method for higher ploidy.

To evaluate HapTree's performance, we consider the probability that HapTree finds the exact solution, as well as compute the *vector error* of a proposed solution, a scoring mechanism for when the exact solution is known, which we newly define to generalize the commonly-used switch error to genomes of higher ploidy. In addition, for triploid genomes, we demonstrate that our relative likelihood measure significantly outperforms the commonly used minimum error correction (MEC) score [20, 22]; this outperformance becomes even greater as the ploidy increases. Finally, as a proof of concept, we test our method on real diploid sequencing data from NA12878 (1000 Genomes Project) and evaluate the quality of assembled haplotypes with respect to trio-based diplotype annotation as the ground truth. The results indicate that even for diploid genomes, HapTree improves the switch error accuracy within phased haplotype blocks as compared to existing haplotype assembly methods [5], while producing comparable MEC values.

## HapTree-X: RNA-seq

To measure phasing accuracy and assess theoretical accuracy bounds, we define *concordant expression* to be when the DASE of a SNP agrees with the DHE of the gene to which the SNP belongs; that is when the majority allele (allele present in the majority of the reads overlapping the SNP locus) is in agreement with the expected majority allele as determined by the DHE. We show that under realistic biological assumptions, the solution of maximal likelihood is, intuitively, that which has concordant expression at each SNP locus. Furthermore, we show that the theoretical probability of concordant expression increases exponentially with the coverage level.

We compare the accuracy of phasing (along with the total number of SNPs phased and phased block sizes) DNA-seq and RNA-seq datasets from NA12878 using HapTreeX to that of HapCut [5]. Our results indicate that incorporating DASE information into haplotype phasing increases the total number of SNPs phased, without increasing the switch error rate (with respect to the trio-phased gold-standard annotation). Furthermore, HapTree-X reduces the total number of phased blocks while increasing their overall sizes. Our work shows for the first time that RNA-seq data can be used as a complement to DNA-seq data to improve phasing.

## 5.2 HapTree

In this section, we observe that on simulated polyploid data HapTree substantially improves the phasing capabilities and performance of any existing program. Because real polyploid data is hard to come by, we also evaluate HapTree on real human diploid data and find that, when compared to the more accurate trio-based data as the ground truth [2], HapTree significantly reduces the number of switch errors, while remaining on par in terms of MEC score over existing single-individual haplotype assembly methods for diploid genomes. We also introduce a relative likelihood (RL) score definition for annotation-free evaluation of phasing quality for polyploid haplotype assembly as an alternative to MEC score. Using simulated polyploid sequencing datasets, we demonstrate that RL-score performs significantly better at capturing haplotype assembly quality than MEC-score as ploidy increases. We will demonstrate these results in the section that follows.

### Scoring and Evaluation

Determining the quality of a phasing solution depends on whether the true phase is known. When no such information is avaliable, the Minimum Error Correction (MEC) score [22] is a widely used scoring function to measure the quality of phasing solutions. The MEC score is defined as the minimum (amongst chromosomes) number of mismatches between a phase $H$ and the read set $R$. A number of existing programs, including HapCut [5], find phasing solutions by optimizing the MEC score in diploid cases. For higher ploidy the MEC score can no longer be reliably used because unlike in the diploid case, the phase of any one chromosome does not determine the phases of the others. Moreover, the MEC score does not distinguish between two separate phases of a pair of SNP loci with different non-zero counts of $(0,0),(0,1),(1,0),(1,1)$ in their vector sets. Finally, unlike in the diploid case, a phase of a pair of SNP loci containing a set of parallel alleles does not prevent it from containing a set of switched alleles as well. To demonstrate these issues, consider two possible vector sets corresponding to phases of a pair of triploid SNPs both with genotype 2: $a : ((0,0),(0,0),(1,1))$ and $b : ((0,0),(0,1),(1,0))\}$. If the read data is $((0,0),(0,0),(0,0))$, it is clear from a probabilistic standpoint that phase $a$ is a better fit, but both $a$ and $b$ have equal MEC scores. This effect is exaggerated as $k$ increases.

When a true phase is available, there are a variety ways to evaluate how accurate any predicted phase is. A widely used measure in diploid phasing is switch error, which is calculated as the number of positions where the two chromosomes of a proposed phase must be switched in order to agree with the true phase. For polyploid phasing, we generalize switch error to *vector error*. In higher ploidy cases, at any SNP locus, it is possible for no chromosomes in a proposed phase to require a switch or anywhere from 2 to $k$ chromosomes to require switches, in order for a proposed phase to agree with the true phase. We do not wish to penalize a solution where only two vectors must be switched at a given position with the same penalty to be used for a solution in which all vectors must be switched. The *vector error* of a proposed phase (with respect to the true phase) is defined by the minimum

number of segments on all chromosomes for which a switch must occur; for the diploid case this score is exactly twice the switch error. One may also think of the vector error as the minimum number of segments a proposed phase and the true phase have in common, less the ploidy. Even for triploid genomes, the vector error is more discriminative than switch error. Consider the following example in Figure 5.1:

| | | | |
|---|---|---|---|
| 1 1 1 0 0 0 1 | 1 1 1 0 0 0 1 | 1 1 1 1 1 0 1 | 1 1 1 0 0 0 0 |
| 1 0 1 0 0 1 0 | 1 0 1 1 1 0 1 | 1 0 1 0 0 0 1 | 1 0 1 1 1 0 1 |
| 0 0 0 1 1 0 1 | 0 0 0 0 0 1 0 | 0 0 0 0 0 1 0 | 0 0 0 0 0 1 1 |
| True Phase | (i): 2 Vector Errors | (ii): 3 Vector Errors | (iii): 4 Vector Errors |

**Figure 5.1:** Examples of *Vector Error* in a sample tetraploid genome; the true phase is on the left and examples with two, three, and four vector errors are on the right.

In Figure 5.1 above, phase (i) is a more accurate phase than (ii), and phase (ii) more accurate than phase (iii). The segments are broken up by row and color: phase (i) having five segments, phase (ii) having six, and phase (iii) having seven. Note that there may be several ways to break a vector set into a minimal number of segments; phase (ii) is such an example. Finally, we remark that vector error can be computed in time $O(kn^2)$, where $k$ is the ploidy and $n$ the block size.

## Relative Likelihood (RL) vs. MEC Simulations

We assessed the effectiveness of our RL score by comparison to MEC score on simulated data. To do so, we simulated reads with error rate 0.02 from a pair of phased $k$-ploid SNP loci for different coverages (5x, 10x, 20x, 100x) and for $k \in \{2, ..., 10\}$. All possible phases were exhaustively enumerated, and phases of the maximal relative likelihood (RL) and phases of the minimal MEC score chosen. We computed the proportion of perfectly phased SNP pairs in both cases (perfect solution rate). Even with two SNP loci, RL significantly outperforms MEC for all $k \geq 3$ (figure 5.2). It is also worth noting that MEC (in comparison to RL) deteriorates more seriously in accuracy as ploidy ($k$) increases (figure 5.2). In addition, we also compared the vector error rate in both cases; for a pair of SNPs, this rate is the number of vectors from the proposed solution that cannot be matched with vectors from the true solution (figure 5.3).

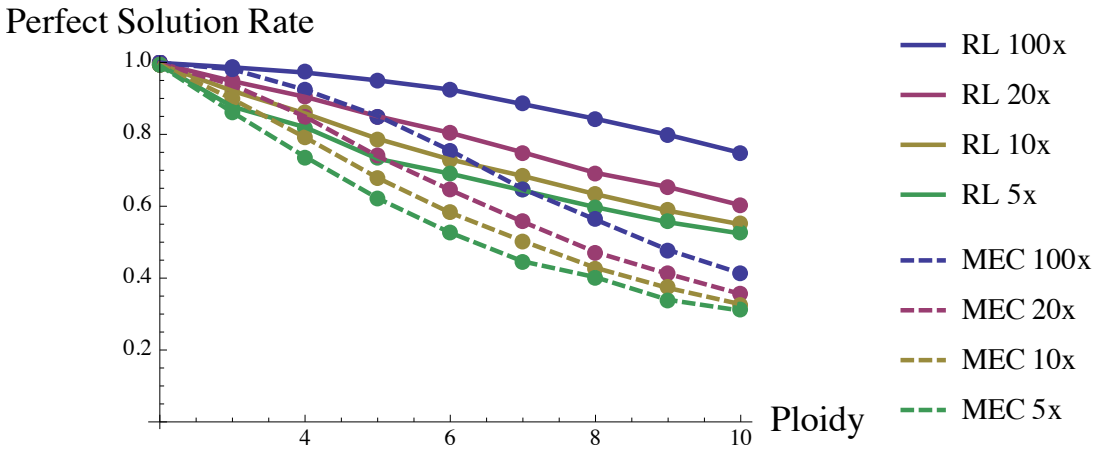


**Figure 5.2:** Proportion of perfectly phased SNP pairs (solid line) and MEC (dashed line) optimization in 10000 trials over 5x, 10x, 20x and 100x coverage.

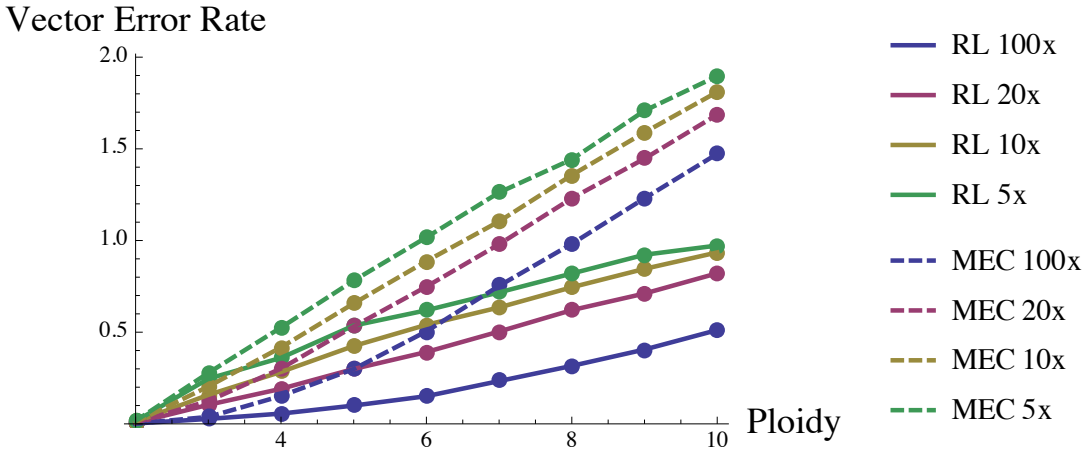


**Figure 5.3:** Vector Error rate for RL (solid line) and MEC (dashed line) optimization in 10000 trials over 5x, 10x, 20x and 100x coverage.

The results demonstrate that the higher the ploidy, the better the relative likelihood (RL) score performs in comparison to MEC score for phasing a pair of SNPs (Figures 5.2 and 5.3). In fact, in simulations where $k \geq 8$, RL with 5x the coverage already outperforms MEC with 100x coverage. For the same coverage, RL always outperforms MEC for $k \geq 3$, and they are equivalent in the diploid case ($k = 2$).

## Comparisons of HapTree and HapCompass on Simulated Polyploid Data

To evaluate the phasing capabilities of HapTree, we compare it with HapCompass [4] (latest version available at: `www.brown.edu/Research/Istrail_Lab/hapcompass.php`), to our knowledge the only other existing program that directly addresses polyploid haplotype assembly, over multiple depth coverage values and component sizes for triploid and tetraploid simulated genomes. We simulated triploid and tetraploid genomes with different block lengths (10, 20 or 40 SNP loci), different coverages (5x, 10x, 20x and 40x), SNP positions, and SNP densities. Throughout the simulations for both the triploid and tetraploid cases, our EXTEND module is run with threshold $\rho = .01$ and PRUNE primarily with threshold $\kappa = .001$. When the current number of haplotype options generated is above 1000, we prune more aggressively with $\kappa = .01$ and when above 5000, with $\kappa = .05$. These parameters are chosen to ensure the efficiency of HapTree by only keeping a tractable collection of promising solutions in each step. We also simulate a read set with uniform error rate and size dependent on coverage. For details about how the reads are simulated, please see **Simulated Polyploid Data Generation**.

We will observe that HapTree consistently out performs HapCompass. The primary reasons for HapTree's superior performance are, first, that HapTree's relative likelihood is more effective than HapCompass's MEC score (see **Relative likelihood vs MEC**); and second, that HapTree's inference algorithm is more accurate than the approximation algorithm used by HapCompass.

## Run-time evaluation

Not only does HapTree outperform HapCompass on phasing quality, it is also significantly faster, especially for longer block length. The median runtimes for block length 10 and 10x coverage were $(0.00702, 0.633)$ seconds for HapTree and HapCompass, respectively; for block length of 40 and 40x coverage, they were $(0.0279, 13.099)$ seconds, respectively.

## Triploid Simulation Results

For the triploid case, we observed that HapTree finds a perfect solution at a rate independent of the number of SNPs used in the simulation; in contrast, HapCompass declines in performance the larger the block size (figures 5.4 and 5.5). While both HapTree and HapCompass improve steadily the higher the coverage, in every case HapTree significantly outperforms HapCompass; the least significant improvement of 63% occurs in the case of 10 SNP loci and 10x coverage, whereas the most significant improvement occurs in the case of 40 SNP loci and 40x coverage. For both vector error rate and likelihood of perfect solution, we find that HapTree substantially outperforms HapCompass.

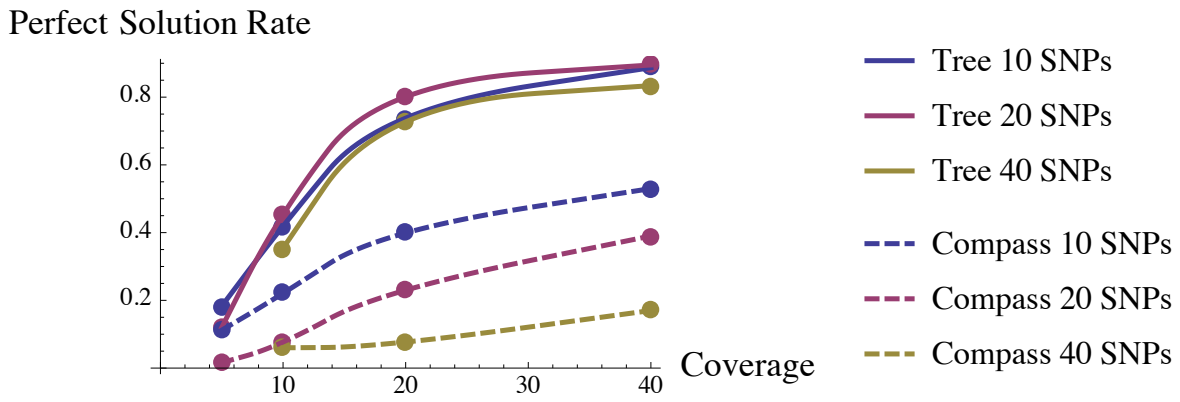


**Figure 5.4:** HapTree (solid lines) and HapCompass (dashed lines) on simulated triploid genomes: Likelihood of Perfect Solution, 1000 Trials, Block lengths: 10, 20, and 40.

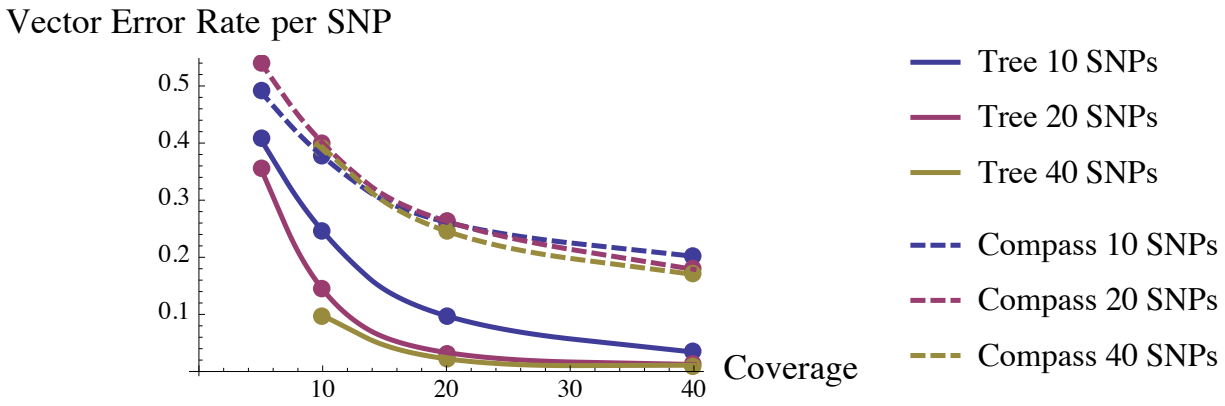


**Figure 5.5:** HapTree (solid lines) and HapCompass (dashed lines) on simulated triploid genomes: Vector Error Rates, 1000 Trials, Block lengths: 10, 20, and 40.

## Tetraploid Simulation Results

For tetraploid simulations, HapTree significantly outperforms HapCompass with block length of 10 SNP loci (Figures 5.6 and 5.7). For larger block lengths HapCompass arrives at the perfect solution at a rate of less than 1%; HapTree however does so at a rate between 40% and 70% depending on block size and coverage at least 20x.

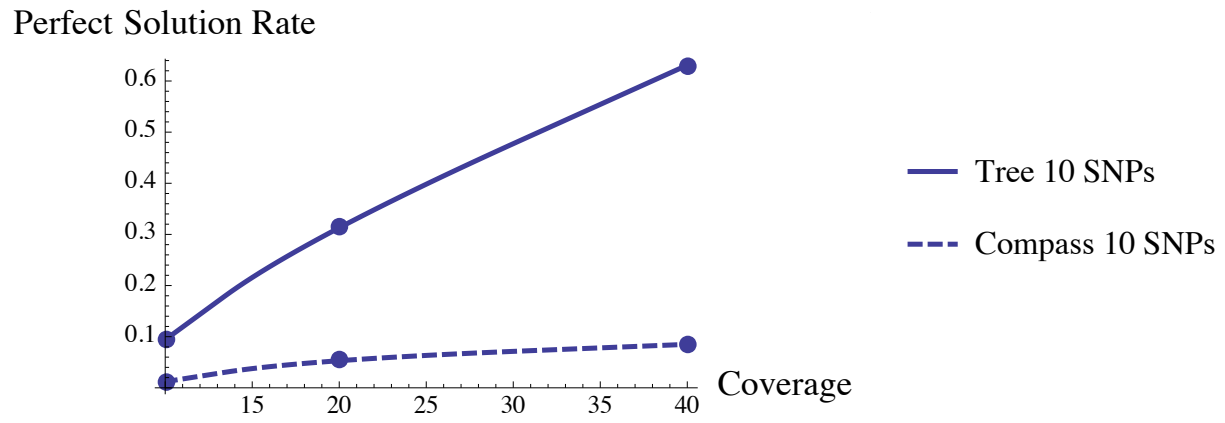


**Figure 5.6:** HapTree (solid line) and HapCompass (dashed line) on simulated tetraploid genomes: Likelihood of Perfect Solution, 1000 Trials, Block length: 10.

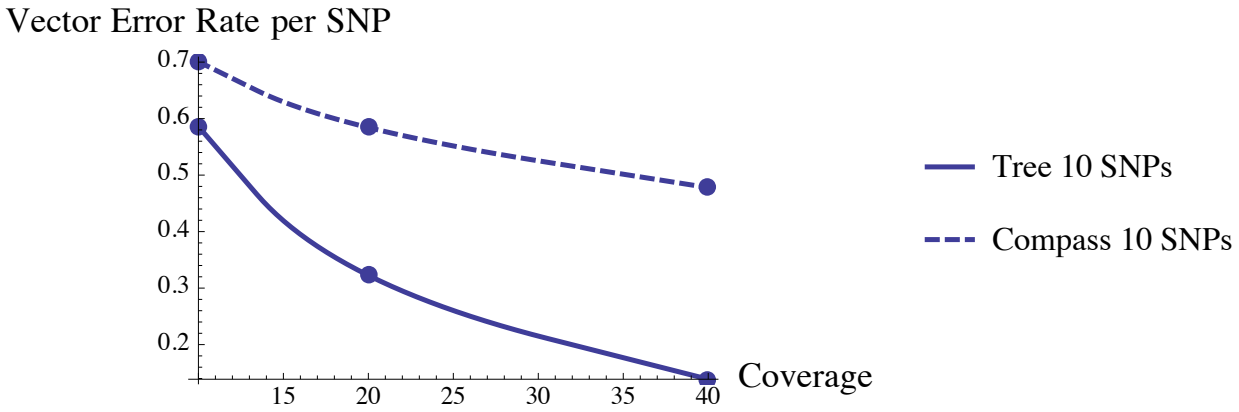


**Figure 5.7:** HapTree (solid line) and HapCompass (dashed line) on simulated tetraploid genomes: Vector Error Rates, 1000 Trials, Block length: 10.

## Performance for Varied Allele Error Rates

We varied the allele error rates (.001, .02, .05, and .1) and observed decreases in accuracy that vary approximately linearly with the (uniform) allele error rates (Figure 5.9). The allele error rate is the likelihood of the sequencing technology to report the incorrect allele for a given position in one read. We ran 10000 trials for simulated triploid genomes of block size 10, with coverages 10x, 20x, and 40x.

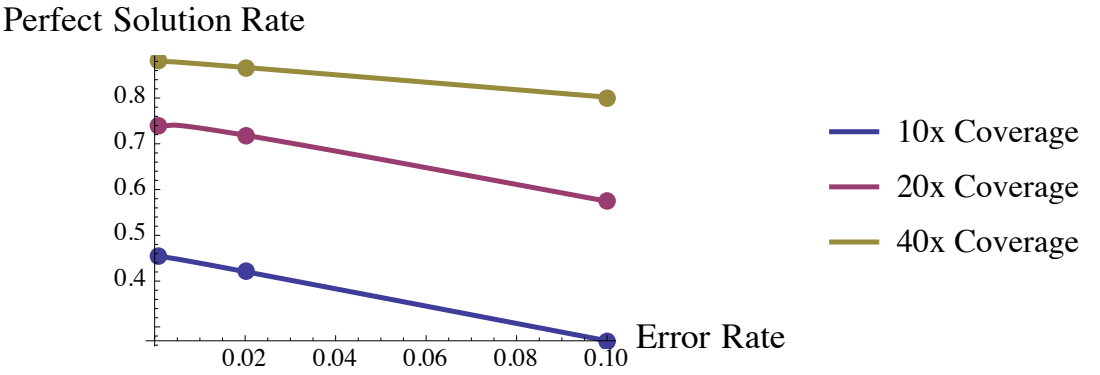


**Figure 5.8:** HapTree performance over varied error rates (.001, .02, .05, .1) and coverages (10x, 20x, 40x) on simulated triploid genomes: Likelihood of Perfect Solution, 10000 Trials, Block length: 10.

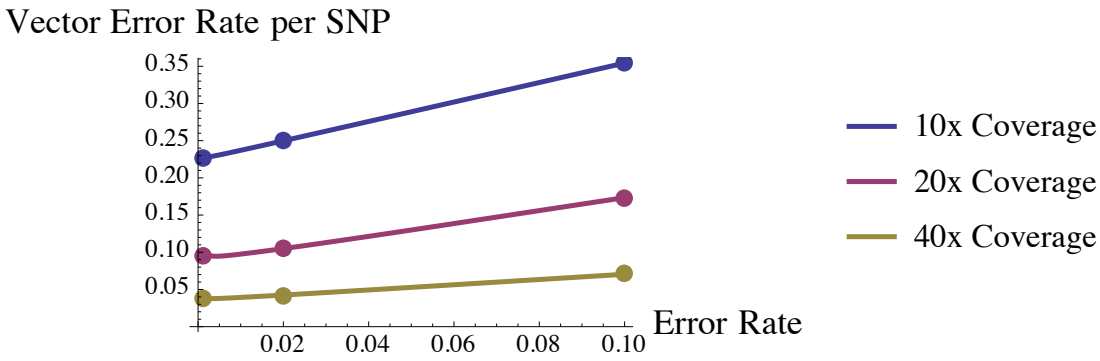


**Figure 5.9:** HapTree performance over varied error rates (.001, .02, .05, .1) and coverages (10x, 20x, 40x) on simulated triploid genomes: Vector Error Rates, 10000 Trials, Block length: 10.

For the simulations above in Figure 5.9, we modeled our read data on Illumina sequencing technologies; for more details, please see section **Simulated Polyploid Data Generation** below. We also ran simulations on longer read data, modeled after 454 sequencing technologies and found almost identical results.

## Results on Real Diploid Data

As seen in the results of Geraci et al. [16], there is no perfect solution for diploid phasing. HapCUT is one of the methods reported that consistently performs best or close-to-best for a variety of experiments. For a proof of concept of how HapTree would perform on real data, we ran HapTree and HapCUT using 454 and Illumina sequencing data of the well-studied NA12878 genome (1000 Genomes Project Phase 1) [2], and compared MEC scores as well as switch errors to a trio phasing annotation accepted as ground truth. The trio phasing annotation represents a high quality diplotype of NA12878 for all SNP sites where either parent (NA12891 or NA12892) is homozygous [2]. Note that we computed the number of switch errors within connected SNP components only, against SNPs whose phase has been determined by the trio-based phasing; we then sum over components. In this case, HapTree was run with a uniform error rate of .02, an EXTEND threshold .001, and primarily with a PRUNE threshold of .001. We begin to prune more aggressively when we have at least $100, 500$, or 1000 possible haplotypes with thresholds of $(.01, 05, .1)$ respectively. For the vector set prior, from examining the read data, we ran HapTree with parallel bias $p = .8$.

We found that HapTree and HapCUT perform almost identically in MEC scores, with HapCUT having marginally smaller scores for both 454 and Illumina data sets. It is worth noting that HapCUT optimizes MEC score, and MEC score measures only the consistency between a phasing solution and read data, not with the true phase.

Notably, when comparing to the ground-truth phase as determined by trio-based phasing, we found HapTree significantly outperforms HapCut in terms of switch error rate for the phasing experiments on the NA12878 genome for 454 and Illumina datasets. Although our method is not primarily designed for phasing diploid genomes, it is still able to achieve better phasing results, when compared to the state-of-the-art diploid method. Again, the results on real-world read datasets showed the superiority of our likelihood function over MEC score for NGS-based phasing.

| **Results** | 454 | | Illumina | |
|---|---|---|---|---|
| Method | MEC | Switch | MEC | Switch |
| HapTree | 32818 | **2978** | 20339 | **1888** |
| HapCut | 32781 | 3192 | 20290 | 1933 |

**Table 5.1:** Results of switch error (switch) and MEC score for HapTree and HapCUT of whole-genome phasing using 454 and Illumina data.

## Simulated Polyploid Data Generation

**Reads.** To generate a paired-end read, we uniformly choose a starting point on the genome (we make sure the genome starts sufficiently before the first SNP and ends at the last). We fix the read-end length (read_len) to be 150. The fragment length (frag_len) is normally

distributed with a mean of 550 and standard deviation of 30, but with min and max lengths of 500 and 600 respectively. The insert length (insert_len) is determined by the fragment length and read-end length, that is, insert_len = frag_len - 2read_len. Once we know the start and fragment length, we must choose from which chromosome to read; we do so uniformly from the $k$ chromosomes. Finally, we add uniform error to the read; we choose a rate of .02, based on the reported error rate of Illumina sequencing technologies. For every SNP that the read covers, independently with probability $\epsilon$ we flip the allele to any other allele; two-thirds of the time when we have this error, we can see that the allele present is neither the reference nor the alternative, and therefore we delete it. Hence, conditional on seeing a SNP in a read, it is incorrect with probability $\varepsilon = \frac{\epsilon}{1-\frac{2}{3}\epsilon}$ and correct with probability $1-\varepsilon$.

**Genomes.** To simulate a genome, we fix a ploidy ($k$) and the number of SNPs ($n$). We determine the positions for the SNPs by randomly generating the distance between each pair of adjacent SNPs. We do so using a geometric random variable with parameter $p$ (SNP density); this choice is equivalent to assuming that any position is a SNP independently with probability $p$. For phasing purposes, once one has generated the reads, the exact genomic positions are no longer relevant; they were only needed to simulate more accurate read data. We therefore refer to SNPs by their position amongst the SNPs, not their position in the genome. For each SNP, we randomly generate its haplotype, assuming for each chromosome, that the alternative and reference alleles are equally likely; if we generate a homozygous SNP, we try again. This procedure results in the likelihood of genotype $g(s) \in \{1, ..., k-1\}$ equal to $\binom{k}{g(s)}/(2^k-2)$, and all orderings $o \in P_s$ being equally likely. For the simulations discussed we use this model. Note, however, that HapTree is not dependent on this model. When running HapTree on real data, different assumptions ought to be made regarding the distributions of vector sets.

**Coverage.** For any genome, to generate a read set with $C$x coverage we need each base pair to be on average covered by $C$ reads. To determine the number of reads to generate, we must know the length of the genome and the read length (read_len). The expected length of the genome is $\frac{n}{p}$ for SNP density $p$, and the read_len is 150 for each end (of which there are two); therefore we simulate $\frac{Cn}{300p}$ reads for $C$x coverage. Note that many of these reads will see only zero or one SNP(s), thus for $C$x coverage the number of useful reads for any SNP will be less than $C$.

## 5.3 HapTree-X

### Theoretical Performance

We demonstrate in section 3.10 the differential haplotypic expression level of a gene, $\beta$, and its coverage determine likelihood of concordant expression. We show this relationship below for varying $\beta$ and levels of coverage. While these functions are derived from an idealized model of the data (for genes without alternative splicing and no amplification bias), this relationship suggests that as the depth-coverage of a dataset increases, so does the likelihood of concordant expression, and hence the accuracy of HapTree-X.

Figure 5.10 displays the theoretical curves depicting the exponential growth of likelihood of concordant expression as a function of coverage and $\beta$, as described in section 3.10. We infer from this theoretical result that requiring a lower bound of DHE is beneficial for reliable DASE-based phasing given moderate coverage (30-50+).

Furthermore, we present a table including minimum coverage required to obtain a probability of at least $1 - 10^{-\alpha}$ of concordant expression, given $\beta$.

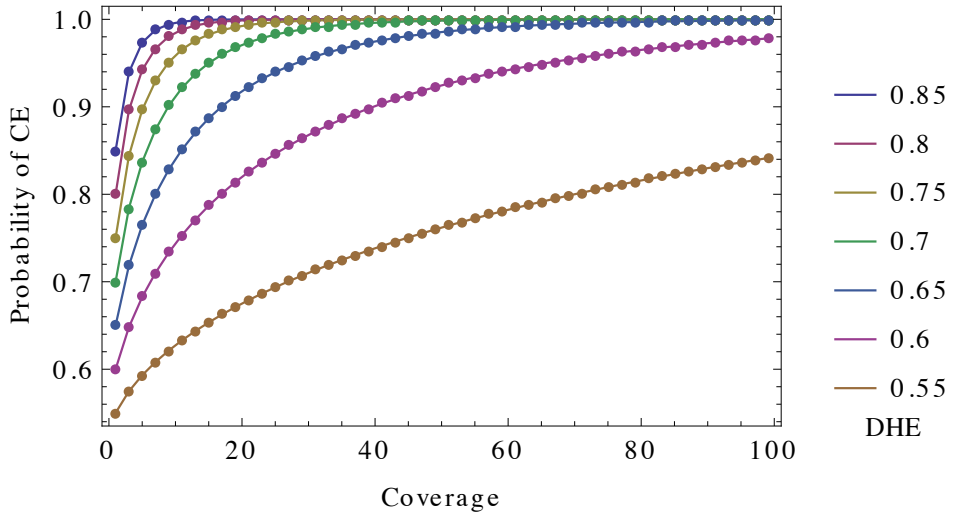


**Figure 5.10:** Likelihood of concordant expression (CE) as a function of coverage and differential haplotypic expression $\beta \in [.55, .6, .65, .7, .75, .8, .85]$

| $\beta \setminus \alpha$ | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| .85 | 9 | 15 | 21 | 27 |
| .8 | 13 | 21 | 31 | 41 |
| .75 | 19 | 33 | 49 | 63 |
| .7 | 31 | 55 | 79 | 105 |
| .65 | 57 | 101 | 147 | 193 |
| .6 | 133 | 235 | 339 | 447 |
| .55 | 539 | 951 | 1377 | 1811 |

**Figure 5.11:** Coverage needed to obtain likelihood $1 - 10^{-\alpha}$ of concordant expression given a differential haplotypic expression of $\beta$ and an assumed opposite allele error rate of 2%.

## Experimental Results

Reported RNA-seq phasing results using HapTree-X for a well annotated human lymphoblastoid cell line (GM12878) provide strong evidence for long-distance haplotype phasing capability of paired-end RNA-seq read alignments as well as the use of differential allele-specific expression as a practical haplotype reconstruction tool. Used jointly with genome reads in genotyping studies, RNA-seq reads can provide long distance scaffolds in order to be used for extending and merging haplotypes inferred from genome reads as well as introducing new long-distance phasing instances not possible to attain using short genome sequencing reads. We observe that compared to a state-of-the-art sequence-based haplotype reconstruction method, HapCut [5], HapTree-X, increases the total number of SNPs phased along with the sizes of phased haplotype blocks with improved accuracy, leveraging RNA-seq reads that only cover a single heterozygous-SNP in the transcriptome.

### Datasets and Experimental Setup

We evaluate haplotype reconstruction performance of HapTree-X on diploid RNA-seq and DNA-seq read datasets from GM12878, a well-studied lymphoblastoid cell line from a human female individual with European Ancestry (1000 Genomes Project [10]).

To assess the accuracy of phased haplotype blocks generated by HapTree-X, we compare our phasing results to a high-quality trio-phased SNP annotation of GM12878 (1000 Genomes Project Phase I), the gold-standard phasing reference. RNA-seq raw read datasets of GM12878 are obtained from ENCODE CSHL Long RNA-seq (wgEncodeCshlLongRnaSeq) [28] track with average sequencing depth of 100 million mate-pairs (2x76bp), transcriptome fragments sequenced from the nucleus with Poly-$A^+$ and Poly-$A^-$ profiling.

For each RNA-seq dataset, we performed 2-pass alignments using STAR aligner v2.4.0d [13] by initially aligning raw reads to hg19 reference genome and then realigning reads to a

second index generated from the splice junctions inferred from the first alignment.

We restricted DASE-based phasing within HapTree-X only to the SNPs that are located within the same gene in the GENCODE gene annotation v19 (wgEncodeGencodeCompV19) [17]. For joint DNA-seq and RNA-seq phasing experiments, we obtained genome sequencing reads of NA12878 from 1000 Genomes Project Pilot 2 release aligned to the hg19 reference genome using bwa aligner [21] and input both genome and transcriptome reads to the HapTree-X haplotype reconstruction framework.

We compared our results to those of HapCUT v0.7 sequence-based haplotype reconstruction tool [5]. To accommodate for long range splicing-junctions within RNA-seq read alignments, we defined maximum insert-size (maxIS) parameter to be longer than each chromosome's length.

### Results from GM12878 data sets

In the RNA-seq read datasets from GM12878 (PolyA+ and PolyA- together), we observe that majority of the reads ($\sim$ 89%) only cover a single heterozygous SNP in the genome. The distribution of read sizes are given in Table 5.2. Of the 19782889 reads containing one SNP, we are able to confidently assign expression biases to 675892 of them; we use these reads to phase, increasing the total number of reads to be used in phasing by 28%.

| **Read Size** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | $8-13$ |
|---|---|---|---|---|---|---|---|---|
| **Count** | 19782889 | 2027207 | 290489 | 47424 | 17176 | 11941 | 10623 | 9119 |
| % | 89.12 | 9.133 | 1.311 | .2137 | .0774 | .0538 | .0479 | .0411 |

**Table 5.2:** Distribution of read sizes (#heterozygous-SNPs covered) in GM12878 RNA-seq data (PolyA+ and PolyA-).

Table 5.3 summarizes the haplotype reconstruction performance of HapTree-X in comparison to a contig-based algorithm, HapCut. Running HapTree-X without any DASE-based phasing (using only reads covering at least two SNPs) yields identical statistics (besides switch error) to HapCut, as both employ the ReadGraph structure to determine the SNPs and blocks to be phased. The switch error rate of HapTree-X without DASE-based phasing is consistent with that from with DASE-based phasing.

| Datasets \ Stats | SNPs | Switch Errors | Blocks | Edges | SNP Pairs |
|---|---|---|---|---|---|
| HapTree-X (DNA-seq & RNA-seq) | 979181 | 3767 | 298637 | 680544 | 5121692 |
| HapCut (DNA-seq & RNA-seq) | 978811 | 5718 | 298710 | 680101 | 5101488 |
| HapTree-X (RNA-seq) | 220849 | 641 | 88355 | 132494 | 412534 |
| HapCut (RNA-seq) | 220386 | 669 | 88403 | 131985 | 380718 |
| HapTree-X (DASE only) | 1580 | 6 | 435 | 1145 | 4884 |

**Table 5.3:** Haplotype reconstruction results from HapTree-X and HapCut using DNA-seq and RNA-seq datasets from NA12878. Both HapCut and HapTree-X results are reported on RNA-seq read datasets as well as DNA-seq and RNA-seq merged datasets. DASE-based phasing only results from HapTree-X are also reported. For each dataset we report total number of phased SNPs, switch errors, haplotype blocks, edges and SNP pairs.

Results indicate that incorporating differential allele-specific expression in haplotype phasing increases the total number of SNPs phased, without increasing the switch error rate (with respect to the trio-phased gold-standard annotation). Furthermore, HapTree-X reduces the total number of blocks while increasing their overall sizes. We represent this by #Edges = #SNPs − #Blocks, equivalently the total number of pairs of adjacent (within a block) phased heterozygous-SNPs. This is also demonstrated by the large increase of total phased SNP pairs (any two SNPs within the same block). This indicates that HapTree-X produces longer haplotype blocks as a result of DASE-based phasing, as desired.

As discussed in section 3.4, the solution of maximum likelihood (for any gene $g$) corresponds to that with concordant expression at all SNP loci within $g$. HapTree-X therefore uses a threshold $\lambda$ (negative log-likelihood of concordant expression) which requires any SNP to be concordantly expressed with probability at least $1 - e^{-\lambda}$, in order to be phased. We run HapTree-X while varying this threshold $\lambda$; we compute the percentage of concordantly expressed SNPs and the total phased SNPs as we increase this threshold. As the threshold increases, HapTree-X demands any SNP to be phased to have a correspondingly high likelihood of concordant expression; as a result, the phasing accuracy of HapTree-X increases. The cost paid for this increase in accuracy is a decrease in the total number of SNPs phased, as seen in Figure 5.12.

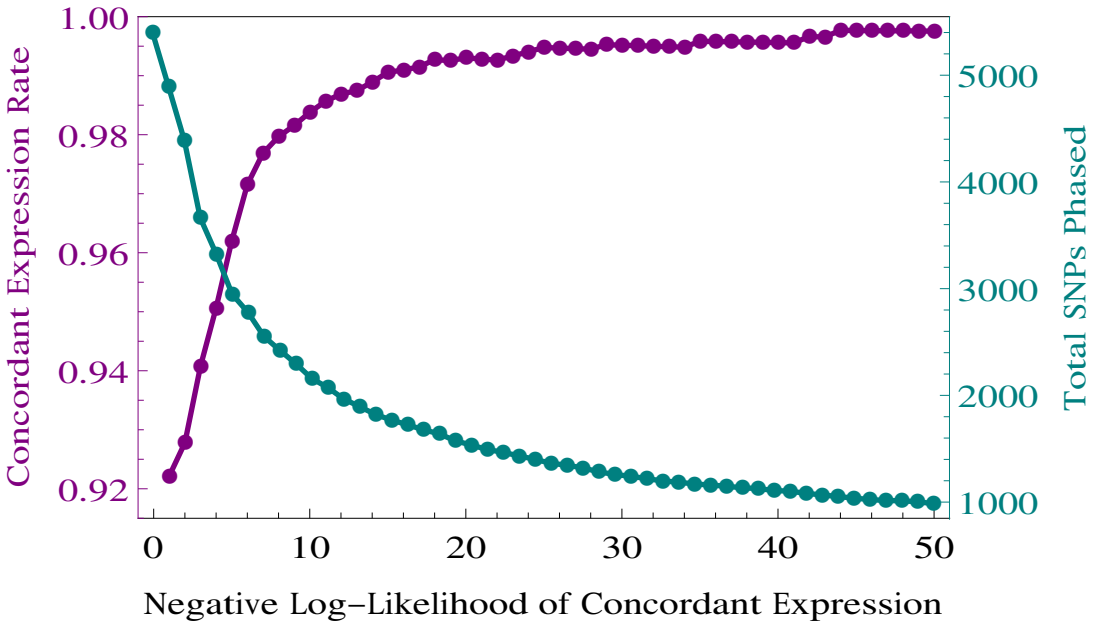


**Figure 5.12:** Rate of concordantly expressed SNPs (purple) and total number of SNPs phased (green) by HapTree-X, as a function of $\lambda$, the negative log-likelihood of concordant expression.

For the results reported in Table 5.3, we used a threshold value of 20. In theory, this threshold value $\lambda$ would produce a percentage of concordantly expressed SNPs equal to $1 - e^{-\lambda}$; however because of the structural noise commonly observed in aligned RNA-seq data due to false mapping, RNA-editing, as well complex alternative splicing events, we require a $\lambda' > \lambda$ to meet desired accuracy levels. Additionally, we require an estimated $\beta \geq .6$ for any gene to be phased using DASE, for motivation see Figure 5.10. Finally, we have several methods for managing alternative splicing events. HapTree-X can (1) avoid all genes with alternative splicing, (2) phase $s, s'$ only if the set of isoforms containing $s, s'$ are equal, and (3) phase independent of isoforms but require $s, s'$ to have coverage and DASE that are sufficiently similar. (3) was used in Table 5.3; (2), and especially (1), result in higher accuracy for lower $\lambda$, but of course phase fewer SNPs.

# Chapter 6

# Phasing with Multiple Isoforms

We have seen, with HapTree-X, that it is possible to perform phasing based on differential haplotypic expression (DHE) by making use of differential allele specific expression (DASE). In the case of HapTree-X, we restricted phasing depending on the structure of the isoforms of a gene; phasing in the general case of multiple isoforms is much more complicated. We discuss some cases in which phasing when there are multiple isoforms can be achieved in this chapter.

Phasing based on DHE and DASE is immediately made more complex by the existence of multiple isoforms. This is primarily because the rates of differential haplotypic expression are independent across isoforms, and furthermore, each isoform is expressed at a different frequency. Fortunately, methods for determining the relative frequencies of expression of isoforms, or *quantification*, have been studied in [27, 23, 26]. We therefore, in this thesis, will assume that these relative frequencies are known.

Not only are isoforms expressed at different frequencies, but the two haplotypes of any isoform may be differentially expressed as well; it is this differential expression that we wish to use for haplotype reconstruction. Unfortunately, these DHE rates are not known and we will see that depending on the situation they may or may not be able to be determined. Furthermore, we will see that knowing the differential expressions may not always be enough to determine the haplotypes, and vice versa.

## 6.1 Definitions and Notation

Recall the goal of phasing is to recover the unknown haplotypes, $H = (H_0, H_1)$, which contain the sequence of variant alleles inherited from each parent of the individual. From now on, we will restrict ourselves to one gene, $G$, with isoforms $\{I_1, ..., I_k\}$. Let $S_i$ denote the set of heterozygous SNPs that are present within the exons of the isoform $I_i$; we let $S = \cup_i S_i$. For any $s \in S$, we let $H_0[s], H_1[s]$ denote the allele present at $s$ on the haplotypes respectively.

Suppose $|S| = n$ and $S = \{s_1, ..., s_n\}$, we can define the *Isoform Matrix* $M$ (an incidence matrix), whose rows correspond to the isoforms $I_i$ and columns correspond to the set of

SNPs $S$; that is $M = \{m_{i,j}\}$ where

$$m_{i,j} = \begin{cases} 1 & \text{if } s_j \in S_i \\ 0 & \text{if } s_j \notin S_i \end{cases}$$

We assume that the quantification of the isoforms is known; that is each isoform $I_i$ is expressed at a rate proportional to $\alpha_i$ For each SNP $s_j$, we define the relative SNP expression level as $A_j$, where

$$A_j = \sum_{i:s_j \in I_i} \alpha_i.$$

Let $\beta_i^M$ and $\beta_i^P$ (with $\beta_i^M + \beta_i^P = 1$) denote the expressions of the maternal and paternal haplotypes respectively for isoform $I_i$; let $\delta_i = \frac{\beta_i^M - \beta_i^P}{2}$.

We assume that $M$ has all distinct rows, that is: isoforms covering identical sets of SNPs are considered to be the same. If this is not the case, say $I'_1, ..., I'_{k'}$ are all identical with quantifications $q'_1, ..., q'_{k'}$ and differential expressions $\delta'_1, ..., \delta'_{k'}$, then we may represent these isoforms as one isoform $I'$ covering the same set of SNPs with quantification $q' = \sum q'_i$ and differential expression $\delta' = \sum q'_i \delta'_i$.

Finally, we let

$$\sigma_j = \begin{cases} 1 & \text{if } H_0[s_j] = 0 \\ -1 & \text{if } H_0[s_j] = 1 \end{cases}$$

Suppose we are given, for each SNP $s_j$, the proportion of alleles expressed that are the reference allele (0) and alternative allele (1); we denote these as $v_j^0$ and $v_j^1$, respectively. In our ideal model of the world, all isoforms are known exactly, each isoform is expressed infinitely (in proportion to its quantification), no read covers more than one SNP, and differential expression is exactly constant across an isoform. In this case, the following should hold:

$$v_j^0 = \frac{1}{A_j} \sum_{i=1}^{k} m_{i,j} \alpha_i (\frac{1}{2} + \sigma_j \delta_i) \tag{6.1}$$

$$v_j^1 = \frac{1}{A_j} \sum_{i=1}^{k} m_{i,j} \alpha_i (\frac{1}{2} - \sigma_j \delta_i) \tag{6.2}$$

## Matrix Representation

It will be useful to formulate these relations in terms of matrices. Let $Q = \{q_{i,j}\}$ be the $k \times k$ diagonal matrix with entries corresponding to the quantifications levels, that is: $q_{i,i} = \alpha_i$. Let $A = \{a_{i,j}\}$ be the $n \times n$ diagonal matrix containing the relative SNP expression levels $A_j$, that is $a_{j,j} = A_j$. We can then define the *Weighted Isoform Matrix*, $M' = \{m'_{i,j}\}$, where

$$M' = QMA^{-1}.$$

By definition $m'_{i,j} = \frac{\alpha_i m_{i,j}}{A_j}$.

The relations described in 6.1 and 6.5 can be represented equivalently as matrix relations as well. To do so, we define $\delta = (\delta_1, ..., \delta_k)$ (with $-\frac{1}{2} \leq \delta_i \leq \frac{1}{2}$) and let $S = \{s_{i,j}\}$ be the $n \times n$ diagonal matrix containing the $\sigma_j$, with $s_{j,j} = \sigma_j$. Finally, let $v^0 = (v^0_1, v^0_2, ..., v^0_n)$ (and let $v^1$ be defined analogously), then 6.1 may be expressed as

$$v^0 = f_k M' + \delta M' S$$

$$v^1 = f_k M' - \delta M' S$$

where $f_k = (\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2})$ and is of length $k$. Since the columns of $M'$ each sum to one, $f_n = f_k M'$.

Let $v = (v^0, v^1)$, since $v^0 + v^1 = (1, ..., 1)$, without loss of generality will restrict ourselves to $v^0$. Let

$$z = v^0 - f_n = \delta M' S \tag{6.3}$$

$$z' = (v^0 - f_n) S^{-1} = \delta M'. \tag{6.4}$$

$S$ is invertible as $\sigma_j = \pm 1$; indeed $S = S^{-1}$.

We will see that being able to phase with multiple isoforms is equivalent to the existence of unique (up to sign) $S$ such that there exists $|\delta_i| \leq \frac{1}{2}$ where $S, \delta$ satisfies 6.3.

## 6.2 Problem Statement

We assume an idealized model of the world, where all isoforms of a gene are perfectly known, as are their quantifications. Furthermore, each isoform is expressed infinitely (in proportion to its known quantification), no read covers more than one SNP, and differential expression is exactly constant across an isoform (though unknown). Given total proportions of reference and alternative alleles being expressed from all isoforms at each SNP locus, we can ask for how many possible pairs of haplotypes, given the known isoforms and their quantifications, it is feasible for us to have observed these allele proportions. In the case where there is a unique pair of haplotypes, we say the haplotypes can be recovered and there exists a unique haplotype solution.

In our notation from the previous section, we assume that the differential expressions $\delta$ and haplotypes $S$ are fixed but unknown. We are given the isoforms $M$, their quantifications $Q$ (and from $M$ and $Q$ we may deduce $A$ and therefore $M'$). Finally we observe $v$, and equivalently $z$. The existence of a unique set of haplotypes is equivalent to there existing a unique (up to sign) $S$ such that there exists $|\delta| \leq \frac{1}{2}$ where $z = \delta M' S$.

**Definition 6.2.1.** *For a given $M'$ and $z$, $S$ is a feasible haplotype solution if there exists $|\delta| \leq \frac{1}{2}$ such that $z = \delta M' S$.*

**Definition 6.2.2.** *For a given $M'$ and $z$, the haplotypes may be recovered if there exists a unique (up to sign) feasible solution $\pm S$.*

## 6.3 Zonotope Representation

We are fortunate to be able to borrow from the rich theory of polytopes, zonotopes, and hyperplane arrangements to describe the set of observations $z$ for which a particular haplotype solution $S$ is feasible. We will state several facts about zonotopes without proof; we refer[1] the reader to [11].

A zonotope may be defined as the Minkowski sum of a set of line segments beginning at the origin. Given a set of points $X \subset V$ (spanning $V$), the zonotope $Z(X) \subset V$ may be written as

$$Z(X) = \Big\{ \sum_{x \in X} t_x x \mid 0 \leq t_x \leq 1 \Big\}.$$

We can shift $Z(X)$, to $Z'(X)$, so that it is centered at the origin, by subtracting the point $\sum_{x \in X} \frac{1}{2} x$ from all points $z \in Z(X)$; equivalently we can write

$$Z'(X) = \Big\{ \sum_{x \in X} t_x x \mid -\frac{1}{2} \leq t_x \leq \frac{1}{2} \Big\}.$$

It is known (but non-trivial to show) that $Z(X)$ (or any polytope) may be written as the intersection over a set of half-spaces each defined by a hyperplane. For the case of zonotopes, there is a nice way to define these hyperplanes, described in Proposition 2.43 of [11]. To do so, let $F$ denote the set of subsets of $X$ that span a codimension one subspace of $V$. For each $f \in F$, let $u_f$ be a linear equation such that $\langle u_f, v \rangle = 0$ defines the subspace spanned by $f$. Finally, define $\mu_f^-$ and $\mu_f^+$ to be the minimum and maximum values attained by $f$ over $Z(X)$, respectively. With these definitions in mind, Proposition 2.43 of [11] states that $Z(X)$ may be written as

$$\bigcap_{f \in F} \big\{ v \in V \mid \mu_f^- \leq \big\langle u_f \,|\, v \big\rangle \leq \mu_f^+ \big\}. \tag{6.5}$$

Let $B^- := \{x \in X \ \langle u_f \,|\, x \rangle < 0\}$ and $B^+ := \{x \in X \ \langle u_f \,|\, x \rangle > 0\}$. In the case of $Z(X)$, $\mu_f^-$ and $\mu_f^+$ are the sums of the values of $u_f$ at $x$, for $x$ in $B^-$ and $B^+$, respectively.

$$\mu_f^- := \sum_{b \in B^-} \langle u_f, b \rangle$$

$$\mu_f^+ := \sum_{b \in B^+} \langle u_f, b \rangle.$$

In the case of the shifted zonotope $Z'(X)$, $u_f$ takes the following minimum and maximum values on $Z'(X)$:

$$\mu_f^- := \sum_{b \in B^-} \frac{1}{2} \langle u_f, b \rangle + \sum_{b \in B^+} -\frac{1}{2} \langle u_f, b \rangle$$

[1] We thank Bryan Gillepsie sharing this reference.

$$\mu_f^+ := \sum_{b \in B^+} \frac{1}{2}\langle u_f, b\rangle + \sum_{b \in B^-} -\frac{1}{2}\langle u_f, b\rangle.$$

We will use this half-space representation to describe the observations $z$ for which a particular haplotype solution $S$ if feasible.

For any fixed $S$, the set of $z$ satisfying 6.3 are those points in the shifted zonotope $Z'(X)$, where $X$ is the set of rows of the matrix $M'S$; we refer to this zonotope as $Z'_S(M')$. Recall $S$ is the diagonal matrix, with entries $\sigma_j = \pm 1$, representing the haplotypes. Without loss of generality, we consider only $Z'(M') = Z'_I(M')$, where $I$ is the identity matrix. We may restrict to this case because any $Z'_S(M)$ is the reflection of $Z'(M)$ through the set of coordinates $j$ such that $\sigma_j = -1$.

Following 6.5, to determine $Z'(M')$, we must compute the linear equations defining the hyperplanes spanned by subsets of rows $M'$ and their minimum and maximum values on $Z'(M')$. To understand the codimension one subspaces spanned by the row vectors of $M'$ it is enough to understand those spanned by the row vectors of $M$ because $M' = QMA^{-1}$, where both $Q$ and $A$ are diagonal. If a subset of rows of $M$ span the subspace defined by

$$\sum_{j=1}^{n} \gamma_j x_j = 0,$$

then the corresponding rows in $M'$ span the subspace defined by

$$\sum_{j=1}^{n} \gamma_j A_j x_j = 0.$$

For fixed $n$, the most general isoform matrix (which we will denote as $\hat{M}(n)$) is the collection of all non-empty binary strings. Determining the linear equations defining all codimension one subspaces spanned by the row vectors of $\hat{M}(n)$ is sufficient for understanding the hyperplane description of any $Z'(M')$, provided $M'$ has at most $n$ columns.

In the following section, we define the hyperplane descriptions of $Z'(M')$ and their corresponding minima and maxima for the case when $n = 2$. The cases of $n = 3$ and $n = 4$ are included at the end of this chapter.

## 6.4 Phasing Two SNPs Given Multiple Isoforms

In the case where a gene $G$ covers exactly two SNPs, we may write down simple necessary and sufficient conditions for the existence of a unique haplotype solution $S$ given $z$, $Q$, and $M$. Recall that we may assume that $M$ has no duplicate rows (that is, isoforms covering identical sets of SNPs are considered to be the same.) Therefore, without loss of generality, we can write:

$$Q = \begin{pmatrix} \alpha_1 & 0 & 0 \\ 0 & \alpha_2 & 0 \\ 0 & 0 & \alpha_3 \end{pmatrix} \qquad M = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{pmatrix} \qquad A = \begin{pmatrix} \alpha_1 + \alpha_3 & 0 \\ 0 & \alpha_1 + \alpha_3 \end{pmatrix}$$

$$M' = QMA = \begin{pmatrix} \frac{\alpha_1}{\alpha_1+\alpha_3} & 0 \\ 0 & \frac{\alpha_2}{\alpha_1+\alpha_3} \\ \frac{\alpha_3}{\alpha_1+\alpha_3} & \frac{\alpha_2}{\alpha_2+\alpha_3} \end{pmatrix}$$

We investigate what kind of solution pairs $\delta, S$ satisfy $zS = \delta M'$ with $\frac{-1}{2} \leq \delta_i \leq \frac{1}{2}$ and $S = \left(\begin{smallmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{smallmatrix}\right)$, with $\sigma_j \in \{\pm 1\}$. The case $S = \pm \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ corresponds to the case in which the two SNPs are *phased in parallel* (the reference allele occurs at both SNP sites on one haplotype and the alternative allele at both SNP sites on the other.) The case $S = \pm \left(\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ corresponds to the SNPs having *switched phase* (one haplotype has a reference allele followed by alternative allele and the other haplotype an alternative allele followed by a reference allele.) Note that if $S, \delta$ is a solution, then $-S, -\delta$ is also a solution and corresponds to relabelling the haplotypes (switching the maternal and paternal haplotypes), and we therefore restrict ourselves to the cases say $S_p = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and $S_s = \left(\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$.

It is possible to determine the phase of the gene $G$ if there do not exist both forms of solution pairs $S_p, \delta$ and $S_s, \delta$; having solutions of both forms imply $z$ can be explained by either parallel or switched phasing, and therefore the phasing is ambiguous. We derive conditions on $z$ which determine when $z$ is such that it is feasible for $G$ to be parallel phased; equivalently when there exist solution pairs $S_p, \delta$ satisfying $zS_p = \delta M'$ and $-\frac{1}{2} \leq \delta_i \leq \frac{1}{2}$. We also derive the corresponding conditions for the case of switched phase. The phasing of $G$ is ambiguous when $z$ satisfies both sets of conditions.

From the viewpoint of section 6.3, we wish to describe $Z'_{S_p}(M')$, its reflection across the $y-$axis, $Z'_{S_s}(M')$, and their intersection; any point $z$ in the intersection $Z'_{S_p}(M') \cap Z'_{S_s}(M')$ implies the phasing of $G$ is ambiguous. In the discussion that follows, we think of $Z'(M')$ as the projection of the cube $\Delta$ defined by $-\frac{1}{2} \leq \delta_i \leq \frac{1}{2}$ by multiplication on the right by $M'$.

To determine conditions for when $z$ can be explained by parallel phasing, we derive equations for the hyperplane description of the zonotope $Z'_{S_p}(M')$ in terms of $z$ and $Q$.

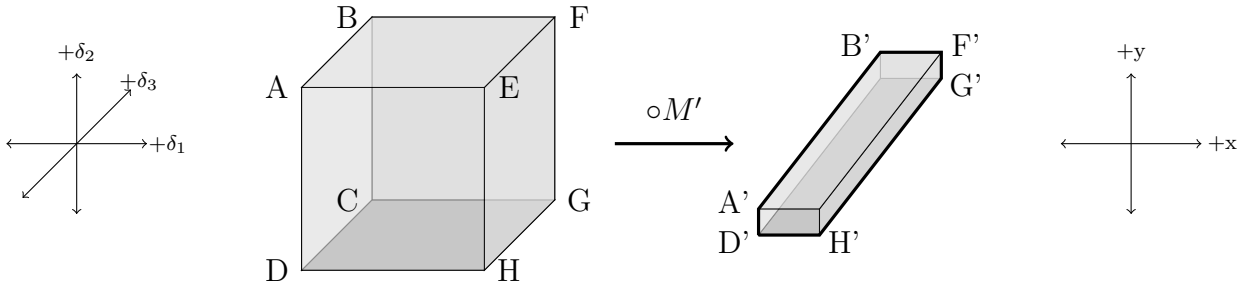


**Figure 6.1:** Projection into 2-space of the cube $\Delta$ under the map: right multiplication by $M'$.

**Proposition 6.4.1.** *The following conditions are necessary and sufficient for the existence of $\delta \in \Delta$ such that $(x, y) = z = \delta M'$, where $\Delta$ is the cube defined by $|\delta_i| \leq \frac{1}{2}$.*

$C_y : |y| \leq \frac{1}{2}$

$C_x : |x| \leq \frac{1}{2}$

$C^-_{xy} : |(\alpha_2 + \alpha_3)y - (\alpha_1 + \alpha_3)x| \leq \frac{\alpha_1+\alpha_2}{2}$

*Proof.* From the discussion in section 6.3, we begin by enumerating the linearly independent subsets of $M'$ who span a codimension 1 subspace, trivially these are the rows $r_1, r_2, r_3$ of $M'$. Equating $(x, y)$ to $\delta M'$ implies the following equations

$$(\alpha_1 + \alpha_3)x = \alpha_1\delta_1 + \alpha_3\delta_3 \tag{6.6}$$

$$(\alpha_2 + \alpha_3)y = \alpha_2\delta_2 + \alpha_3\delta_3 \tag{6.7}$$

$$(\alpha_2 + \alpha_3)y - (\alpha_1 + \alpha_3)x = \alpha_2\delta_2 - \alpha_1\delta_1 \tag{6.8}$$

In general, the coefficient attached to $\delta_i$ on the RHS is what the row $r_i$ evaluates to under the LHS. Each $\delta_i$ is missing from one these equations, and these equations are therefore sufficient for defining the zonotope as an intersection of half-spaces. To determine those half-spaces, we follow 6.3 and minimize and maximize the RHS of each equation, implying the bounds in the statement of Proposition 6.4.1.

□

**Proposition 6.4.2.** *The following conditions are necessary and sufficient for the existence of $\delta \in \Delta$ such that $(x, y) = z = \delta M' \left(\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$, where $\Delta$ is the cube defined by $|\delta_i| \leq \frac{1}{2}$.*

$C_y : |y| \leq \frac{1}{2}$

$C_x : |x| \leq \frac{1}{2}$

$C^+_{xy} : |(\alpha_2 + \alpha_3)y + (\alpha_1 + \alpha_3)x| \leq \frac{\alpha_1+\alpha_2}{2}$

This statement follows from Proposition 6.4.1 by a reflecting across the $y$-axis.

By combining Propositions 6.4.1 and 6.4.2 we can determine when the phasing of $G$ is ambigious; that is, when both $C^-_{xy}$ and $C^+_{xy}$ hold.

We include images depicting the the zonotopes $Z'_{S_p}(M')$ and $Z'_{S_s}(M')$ as determined by 6.4.1 and 6.4.2. As intuition suggests, the larger $\alpha_3$ is to relative to $\alpha_1, \alpha_2$, the smaller the set of $z$ with ambiguous phasing $(Z'_{S_p}(M') \cap Z'_{S_s}(M'))$. This intuition can be explained by the idea that $I_3$ is the only isoform with information about the true phase.

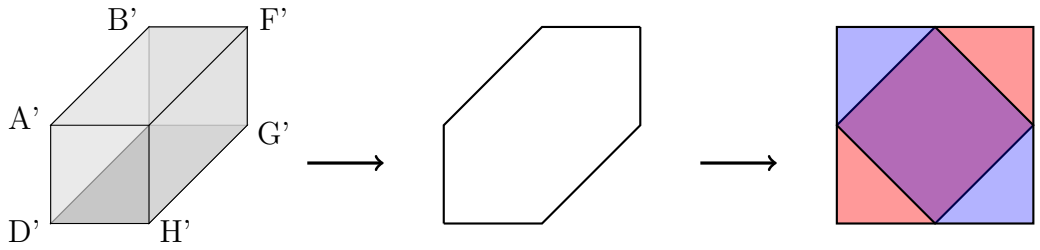


**Figure 6.2:** Feasible regions for parallel phase in red, for switched in blue, and for both in purple with $(\alpha_1, \alpha_2, \alpha_3) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$.

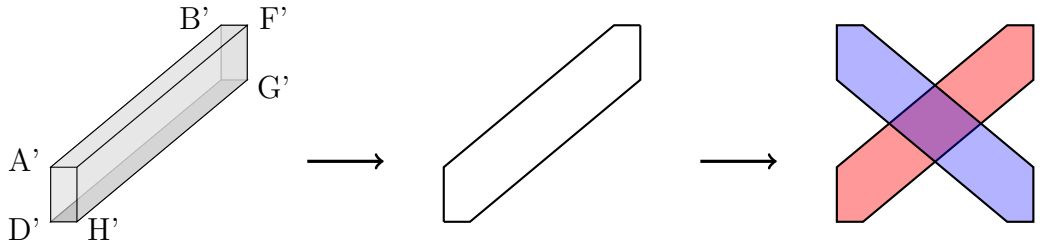


**Figure 6.3:** Feasible regions for parallel phase in red, for switched in blue, and for both in purple with $(\alpha_1, \alpha_2, \alpha_3) = (\frac{2}{20}, \frac{5}{20}, \frac{13}{20})$.

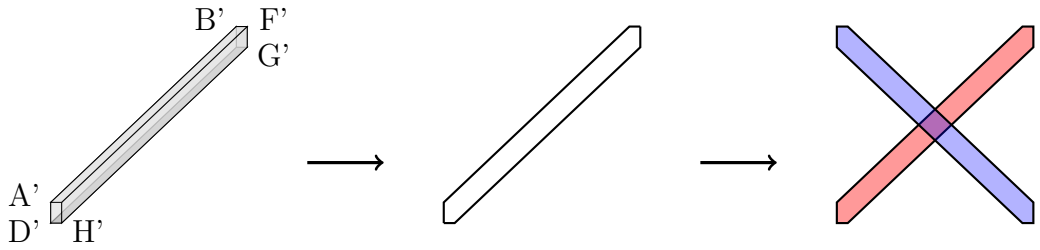


**Figure 6.4:** Feasible regions for parallel phase in red, for switched in blue, and for both in purple with $(\alpha_1, \alpha_2, \alpha_3) = (\frac{1}{20}, \frac{2}{20}, \frac{17}{20})$.

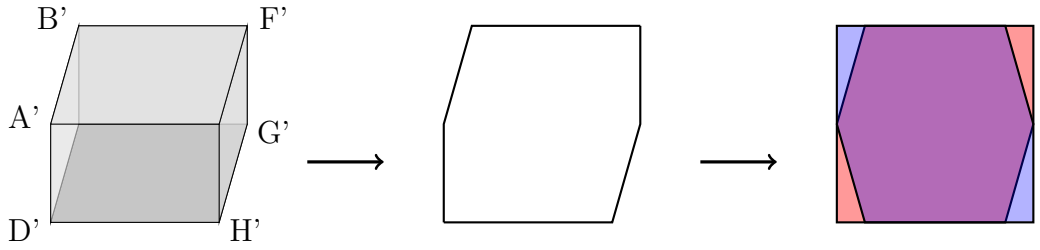


**Figure 6.5:** Feasible regions for parallel phase in red, for switched in blue, and for both in purple with $(\alpha_1, \alpha_2, \alpha_3) = (\frac{3}{4}, \frac{1}{8}, \frac{1}{8})$.

## 6.5 Modeling Noise

The model described in the preceding section assumed a theoretical set up where coverages of isoforms are sent to infinity in proportion to their quantifications, and the differential expression across an isoform is exactly constant. In reality, we do not have infinite coverage. Furthermore, we can have variation of coverage relative to the 'known' quantifications (of an entire isoform or just a SNP) and variation of differential expression within an isoform. These variations can be caused by both random and structural noise. Modeling all of these variations at once is equivalent to finding solutions $\delta, S$ to the equations

$$v_j^0 = \frac{1}{A'_j} \sum_{i=1}^{k} m_{i,j}(\alpha_i + \epsilon_{\alpha,i,j})(\frac{1}{2} + \sigma_j(\delta_i + \epsilon_{\delta,i,j})) \tag{6.9}$$

$$v_j^1 = \frac{1}{A'_j} \sum_{i=1}^{k} m_{i,j}(\alpha_i + \epsilon_{\alpha,i,j})(\frac{1}{2} - \sigma_j(\delta_i + \epsilon_{\delta,i,j})) \tag{6.10}$$

where $A'_j = \sum_{i=1}^{k} m_{i,j}(\alpha_i + \epsilon_{\alpha,i,j})$, such that $|\delta_i| \leq \frac{1}{2}$ and $\sigma_j = \pm 1$, the $\epsilon_\alpha, \epsilon_\delta$ are bounded in some way, and possibly $|\delta_i + \epsilon_{\delta,i,j}| \leq \frac{1}{2}$ as well. Even for fixed $S$, this problem is non-linear since the errors and $\delta$ are unknown. Instead, we discuss the simpler case of modeling variation of differential expression across an isoform.

### Non-Uniform Differential Expression

Assuming fixed and known quantifications, we can model variation at the SNP level of differential expression. Without noise, differential expression ought to be the same for all SNPs across an isoform. We introduce noise limits $\varepsilon = \{\varepsilon_{i,j}\}$ to account for this variation. Given an observation $v^0, v^1$, we say $S$ is a feasible haplotype solution if there exists $|\delta| \leq \frac{1}{2}$ and $\epsilon \leq \varepsilon$ (where $\{\epsilon_{\delta,i,j}\}$) such that the following hold:

$$v_j^0 = \frac{1}{A_j} \sum_{i=1}^{k} m_{i,j}\alpha_i(\frac{1}{2} + \sigma_j(\delta_i + \epsilon_{\delta,i,j}))$$

$$v_j^1 = \frac{1}{A_j} \sum_{i=1}^{k} m_{i,j}\alpha_i(\frac{1}{2} - \sigma_j(\delta_i + \epsilon_{\delta,i,j}))$$

$$z_j = v_j^0 - f_n = \frac{\sigma_j}{A_j} \sum_{i=1}^{k} m_{i,j}\alpha_i(\delta_i + \epsilon_{\delta,i,j}) \tag{6.11}$$

Recall $f_k = (\frac{1}{2}, \frac{1}{2}, ..., \frac{1}{2})$ and is of length $k$ and that the column sums of $M'$ are one.

Furthermore, we may wish to bound $|\delta_i + \epsilon_{\delta,i,j}| \leq \frac{1}{2}$ since those limits correspond to only one haplotype being expressed.

Let $\hat{\epsilon_j}, \hat{\varepsilon_j}$, denote the weighted average of errors and their upper bounds respectively at each SNP, that is

$$\hat{\epsilon_j} = \frac{1}{A_j} \sum_{i=1}^{k} m_{i,j} \alpha_i \epsilon_{\delta,i,j}$$
$$\hat{\varepsilon_j} = \frac{1}{A_j} \sum_{i=1}^{k} m_{i,j} \alpha_i \varepsilon_{i,j}.$$

Setting $\hat{\epsilon} = (\hat{\epsilon_1}, ..., \hat{\epsilon_n})$ and $\hat{\varepsilon} = (\hat{\varepsilon_1}, ..., \hat{\varepsilon_n})$, we may write

$$z = v^0 - f_k = \delta M' S + \hat{\epsilon} S.$$

and therefore

$$|z - \delta M' S| \leq \hat{\varepsilon}. \tag{6.12}$$

If we impose $|\delta_i + \epsilon_{\delta,i,j}| \leq \frac{1}{2}$, then $|z| \leq \frac{1}{2}$ as well.

For fixed $S$ and given $z$, if we insist that $|\delta_i + \epsilon_{\delta,i,j}| \leq \frac{1}{2}$, then the existence of $|\delta| \leq \frac{1}{2}$ satisfying 6.12 does not imply the existence of $|\delta| \leq \frac{1}{2}, \epsilon \leq \varepsilon$ satisfying 6.11. If $|z| \leq \frac{1}{2}$ as it will be in practice, then there will exist a solution where the weighted average over $i$ of $|\delta_i + \epsilon_{\delta,i,j}|$ is less than a half for each $j$.

## 6.6 Experimental Results

We wish to get a sense for whether or not phasing when multiple isoforms are present is possible with the current state of RNA-seq data. To do so, we see how accurate both our idealized and error based models are when implemented on RNA-seq data. We use RNA-seq raw read datasets of GM12878 obtained from ENCODE CSHL Long RNA-seq (wgEncodeCshlLongRnaSeq) [28] track with average sequencing depth of 100 million mate-pairs (2x76bp), and transcriptome fragments sequenced from the nucleus with Poly-A$^+$ and Poly-A$^-$ profiling. For both the Poly-A$^+$ and Poly-A$^-$ profiling nucleus datasets there are two replicates available, yielding a total of four separate datasets. Also obtained from ENCODE CSHL Long RNA-seq are the transcript quantifications [28] based on GENCODE gene annotation v7 [17]. We take as ground truth Illumina's platinum genome VCF file IlluminaPlatinumGenomes_v7.0 [1] phased with both trio and further inheritance constraints.

Based on the gene model mentioned above, we find all *isoform covered* SNP pairs $(s1, s2)$ (that is, those which occur together in an isoform with non-zero quantification; from here on, when we say isoform, we mean an isoform with non-zero quantification.) We can ask what the coverage at $(s1, s2)$ alone says about the feasible phasing solutions of the pair. This problem can be reduced to the setup in section 6.4 by letting $\alpha_1$, $\alpha_2$ and $\alpha_3$ equal the sum of quantifications of all isoforms covering only $s1$, only $s2$, and both $s1$ and $s2$ respectively. Suppose the pair has coverage $[A_1, B_1]$ and $[A_2, B_2]$ where $A_i, B_i$ denote the number of reads

containing the reference allele at $s_i$ and the alternative allele at $s_i$ respectively. In our previous notation, we can write $(x, y) = (\frac{A_1}{A_1+B_1} - \frac{1}{2}, \frac{A_2}{A_2+B_2} - \frac{1}{2})$ and check when either $C^-_{x,y}$ (6.4.1) holds, which would mean parallel phasing is feasible, and when $C^+_{x,y}$ (6.4.2) holds, implying switched phasing is feasible.

$$C^-_{xy}: \ |(\alpha_2 + \alpha_3)y - (\alpha_1 + \alpha_3)x| \leq \frac{\alpha_1 + \alpha_2}{2}$$

$$C^+_{xy}: \ |(\alpha_2 + \alpha_3)y + (\alpha_1 + \alpha_3)x| \leq \frac{\alpha_1 + \alpha_2}{2}$$

In this context, it is possible to have two, one, or zero feasible solutions. In order for phasing with multiple isoforms to be effective, we ideally would like there to exist a unique feasible solution a high fraction of the time, and furthermore for that solution to be the true phasing of the pair. In the table below we count the number of pairs with two, one, and zero feasible solutions. We also report how often the unique feasible solution is accurate (AUFS), by which we mean agrees with the phasing provided in the gold-standard platinum VCF. We also report the same stats in the case where we condition on the coverage being at least 15 for both SNPs; somewhat surprisingly this does not affect AUFS significantly.

| **Results** | PolyA- Rep 1 | | | PolyA- Rep 2 | | |
|---|---|---|---|---|---|---|
| # Feasible Solutions | 2 | 1 | 0 | 2 | 1 | 0 |
| # SNP Pairs | 11695 | 7066 | 26850 | 11928 | 7477 | 27174 |
| % SNP Pairs | 25.64 | 15.49 | 58.87 | 25.81 | 16.05 | 58.34 |
| AUFS Rate | .6514 | | | .6580 | | |
| **Results** | PolyA+ Rep 1 | | | PolyA+ Rep 2 | | |
| # Feasible Solutions | 2 | 1 | 0 | 2 | 1 | 0 |
| # SNP Pairs | 15618 | 8714 | 34682 | 14778 | 8473 | 33561 |
| % SNP Pairs | 26.46 | 14.77 | 58.77 | 26.01 | 14.91 | 59.07 |
| AUFS Rate | .7592 | | | .7285 | | |

**Table 6.1:** Counts and percentage of number of isoform covered SNP pairs with two, one, or zero feasible solutions; accuracy of unique feasible solution (AUFS). No coverage requirement.

| **Results** | PolyA- Rep 1 | | | PolyA- Rep 2 | | |
|---|---|---|---|---|---|---|
| # Feasible Solutions | 2 | 1 | 0 | 2 | 1 | 0 |
| # SNP Pairs | 3891 | 1062 | 6360 | 4019 | 963 | 6189 |
| % SNP Pairs | 34.39 | 09.39 | 56.22 | 35.98 | 08.62 | 55.40 |
| AUFS Rate | .6271 | | | .6449 | | |
| **Results** | PolyA+ Rep 1 | | | PolyA+ Rep 2 | | |
| # Feasible Solutions | 2 | 1 | 0 | 2 | 1 | 0 |
| # SNP Pairs | 10001 | 3086 | 18011 | 9504 | 2881 | 17682 |
| % SNP Pairs | 32.16 | 09.92 | 57.92 | 31.61 | 09.58 | 58.81 |
| AUFS Rate | .7955 | | | .7372 | | |

**Table 6.2:** Counts and percentage of number of isoform covered SNP pairs with two, one, or zero feasible solutions; accuracy of unique feasible solution (AUFS). Coverage at least 15 for all SNPs.

To incorporate error into the model, we follow the formulation in 6.12 and ask, for fixed $\varepsilon$, when there exists a feasible solution to

$$|z' - \delta M' S| \leq \varepsilon. \tag{6.13}$$

Equivalently, for which $z'$ does there exist $z$ with $|z' - z| \leq \varepsilon$ satisfying:

$$z = \delta M' S.$$

Let $z' = (x', y')$, it follows from 6.4.1 that the set of $z'$ satisfying 6.13 are those which satisfy:

$$C^{-}_{xy}(\varepsilon): \ |(\alpha_2 + \alpha_3)y' - (\alpha_1 + \alpha_3)x'| \leq \frac{\alpha_1 + \alpha_2}{2} + (\alpha_1 + \alpha_2 + 2\alpha_3)\varepsilon$$

$$C^{+}_{xy}(\varepsilon): \ |(\alpha_2 + \alpha_3)y' + (\alpha_1 + \alpha_3)x'| \leq \frac{\alpha_1 + \alpha_2}{2} + (\alpha_1 + \alpha_2 + 2\alpha_3)\varepsilon$$

for $S = \pm \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ and $\pm \left(\begin{smallmatrix} -1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$ respectively. Note that $z'$ will always satisfy $C_x$ and $C_y$ by definition.

For any $z' = (x', y')$, we can ask how much error is required in order for some $S$ to be feasible; we let $f_p(x', y')$ and $f_s(x', y')$ denote such an error bound for parallel phased and switched phase solutions respectively:

$$f_p(x', y') = \frac{|(\alpha_2 + \alpha_3)y' - (\alpha_1 + \alpha_3)x'| - \frac{\alpha_1 + \alpha_2}{2}}{(\alpha_1 + \alpha_2 + 2\alpha_3)} \tag{6.14}$$

$$f_s(x', y') = \frac{|(\alpha_2 + \alpha_3)y' + (\alpha_1 + \alpha_3)x'| - \frac{\alpha_1 + \alpha_2}{2}}{(\alpha_1 + \alpha_2 + 2\alpha_3)}. \tag{6.15}$$

We can ask at various error levels, how many isoform covered SNP pairs have two, one, and zero feasible solutions (Figure 6.6). For the SNP pairs admitting only one feasible solution at a given allowed error bound, we compute the proportion of accurately phased SNP pairs (AUFS). As these curves do not vary significantly across Poly-A profile or replicate, we report these results for just Poly-A- replicate 2.

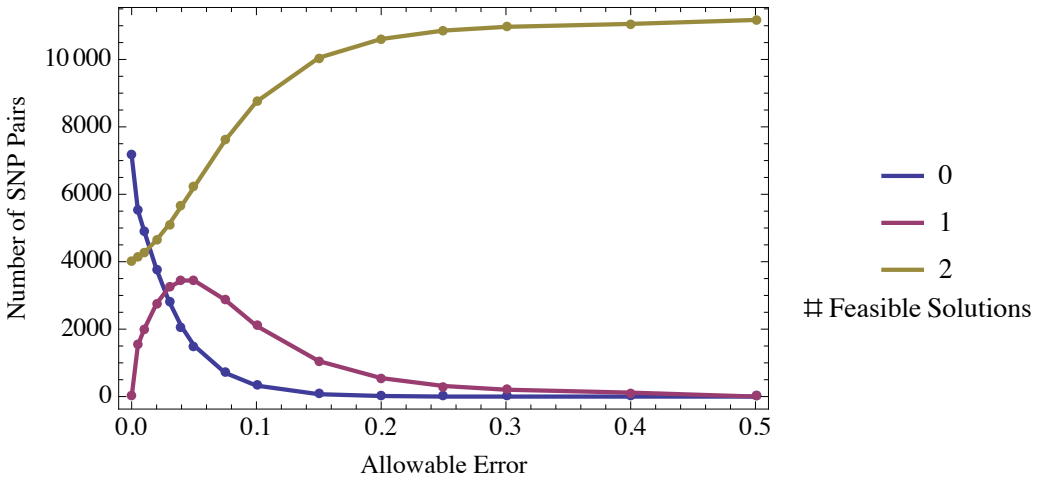


**Figure 6.6:** Counts of SNP pairs with coverage at least 15 with two, one, and zero feasible solutions with varying allowed error for Poly-A- replicate 2.

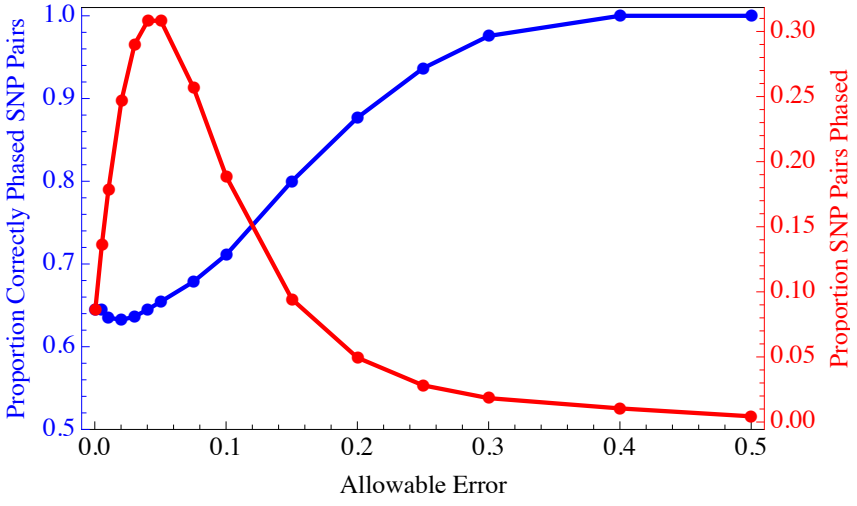


**Figure 6.7:** For varying allowed error, we report the proportion of SNP pairs with a unique feasible solution (red) and the accuracy of that feasible solution (AUFS) (blue) compared against the platinum phased VCF for Poly-A- replicate 2.

Generalizing one step further, we consider phasing a pair of SNPs when one phasing solution requires $\epsilon_m \leq \varepsilon_m$ error in order to be feasible and the other requires $\epsilon_M \geq \varepsilon_M$ error in order to be feasible. Using the notation introduced 6.14 and 6.15, we phase a SNP pair when either $f_p(x', y') \leq \varepsilon_m$ and $f_s(x', y') \geq \varepsilon_M$ or $f_s(x', y') \leq \varepsilon_m$ and $f_p(x', y') \geq \varepsilon_M$. When these conditions are satisfied, we compare the solution with lower error to the true phase. We refer to the upper bound $\varepsilon_m$ as allowable min-error and the lower bound $\varepsilon_M$ as required max-error. By doing so, we attempt to capture the intuition that given one solution which appears to be very good, the worse the alternate solution is, the higher our confidence the good solution is the true solution.

Below we plot how many SNP pairs can be phased for various $\varepsilon_m, \varepsilon_M$ (Figure 6.9) and how often those phasing solutions are accurate (Figure 6.10). Below, each curve corresponds to some $\varepsilon_m \in [0, .02, .05, .1]$. The enlarged points on the curves correspond to $\varepsilon_m = \varepsilon_M$; those points occur in Figure 6.7. In Figure 6.8 for $\varepsilon_m = .05$, we plot the proportion of SNPs phased and phasing accuracy curves jointly. In all cases, we require coverage of at least 15 for all SNPs. As intuition suggests, accuracy appears to be approximately monotonically increasingly, and percentage of SNPs phased approximately monotonically decreasing, differing from the results presented in Figure 6.7.

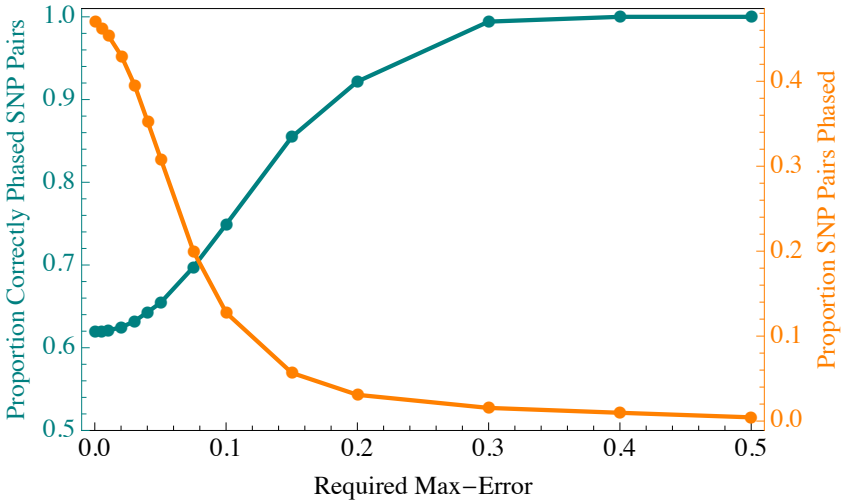


**Figure 6.8:** For allowable min-error = .05 and varying required max-error $\varepsilon_M$, we report the proportion of accurately phased SNP pairs and the proportion of SNP pairs (out of all SNP pairs with coverage at least 15) able to be phased (that is, those with coverage at least 15, min-error $\leq .05$, and max-error $\geq \varepsilon_M$.)

Intuition suggests that with higher coverage, accuracy ought to increase; unfortunately requiring higher coverage limits how often we are able to phase. We fix $\varepsilon_m = .05$ and $\varepsilon_M = .15$ and for varying required coverage, we report what proportion of SNP pairs have

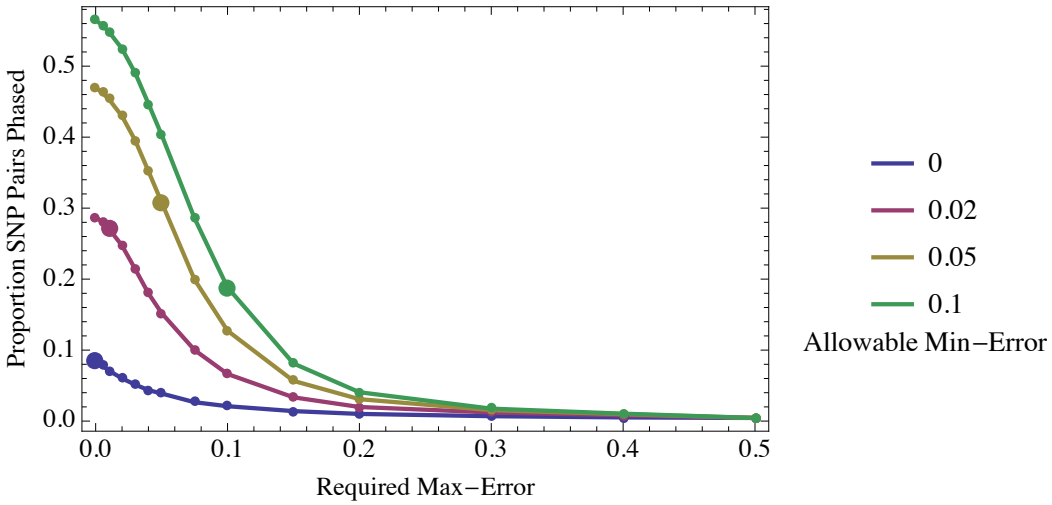


**Figure 6.9:** For allowable min-error $\varepsilon_m \in [0, .02, .05, .1]$ (in purple, magenta, yellow, and green respectively) and varying required max-error $\varepsilon_M$ we report the proportion of SNP pairs (out of all SNP pairs with coverage at least 15) able to be phased: those with coverage at least 15, min-error $\leq \varepsilon_m$, and max-error $\geq \varepsilon_M$.

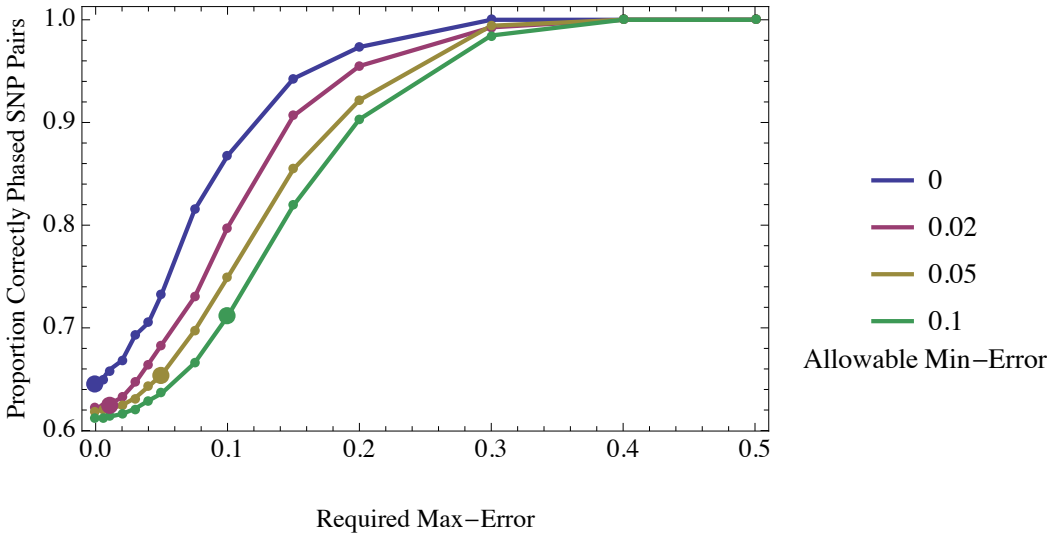


**Figure 6.10:** For allowable min-error $\varepsilon_m \in [0, .02, .05, .1]$ (in purple, magenta, yellow, and green respectively) and varying required max-error $\varepsilon_M$ we report the proportion of SNP pairs that were phased accurately relative to the gold-standard phased platinum vcf. The SNP pairs phased are those with coverage at least 15, min-error $\leq \varepsilon_m$, and max-error $\geq \varepsilon_M$.

the required amount of coverage (Figure 6.11), how accurately we phase when the coverage and error thresholds are satisfied (Figure 6.11), and of SNP pairs satisfying the coverage thresholds, what proportion satisfy the error thresholds and thus we choose to phase (Figure 6.12.)

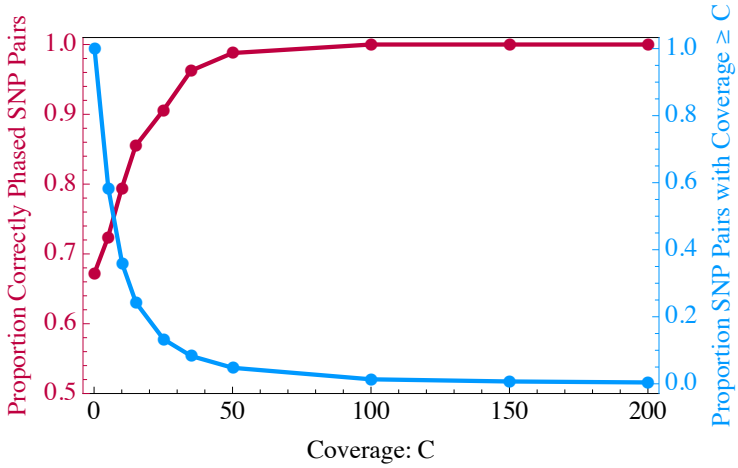


**Figure 6.11:** We report the proportion of SNP pairs with coverage above a varying threshold (blue). Of those pairs, we phase those satisfying the error thresholds: $\varepsilon_m = .05$ and $\varepsilon_M = .15$; we report the proportion of correctly phased SNP pairs (red).

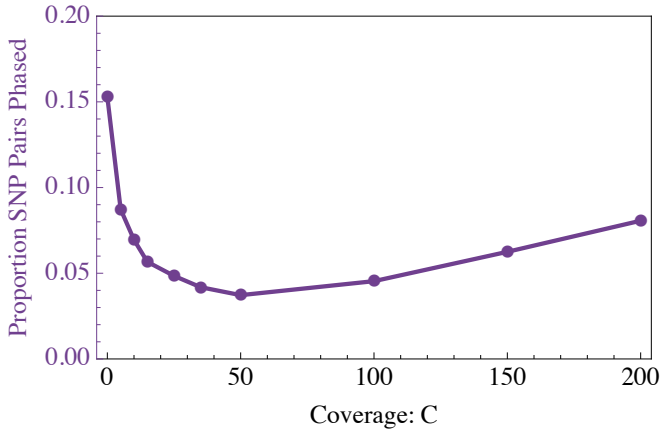


**Figure 6.12:** We report what proportion of SNP pairs with coverage $\geq$ C satisfy the error thresholds $\varepsilon_m = .05$ and $\varepsilon_M = .15$.

## Discussion

We find that attempting to phase SNP pairs in the presence of multiple isoforms without taking into account error leads to low accuracy in phasing results, and furthermore for all replicates and both Poly-A profiles, between $55-60\%$ of SNP pairs have no feasible solutions (Tables 6.1 and 6.2).

If we incorporate allowable error, the percentage of SNP pairs with a unique feasible solution is maximized around an allowable error $\varepsilon = .05$ (Figures 6.6 and 6.7.) As we increase the allowable error, the accuracy rate of phasing increases, but the percentage of SNP pairs with two feasible solutions with goes to 100% (Figure 6.6), as it must, and therefore the percentage with a unique feasible solution goes to zero (Figure 6.7.)

When we allow two dimensional error thresholds, that is, when in order to phase we require one solution be very close to 'perfect' (low required error) and the other very far from it (high required error), we see (unsurprisingly) for fixed allowable min-error, increasing required max-error increases accuracy while decreasing the proportion of SNPs phased (Figures 6.8, 6.9, 6.10). Additionally, we see for fixed $\varepsilon_m$ and $\varepsilon_M$, increasing coverage increases accuracy. In this context, for a fixed pair of error thresholds, the total number of SNP pairs able to be phased decreases as required coverage increases, but the proportion of SNP pairs (out of SNP pairs satisfying the lower bound of required coverage) which satisfy the error thresholds is noisy, but relatively flat when varying coverage.

At this time, it seems that certain pairs of SNPs can be accurately phased in the presence of multiple isoforms: those which satisfy various error and coverage thresholds. It may be useful to incorporate this sort of model into the general HapTree-X framework. At this time, to phase an entire gene of (possibly more than two) SNPs simultaneously, we can employ linear programming for each potential haplotype solution $S$ for the gene to determine whether there exists $z = \delta M'S$ with $|\delta| \leq \frac{1}{2}$, or more generally if there exists a solution based on the error model for non-uniform differential expression in section 6.5. This approach unfortunately involves enumerating $2^{n-1}$ haplotypes, where $n$ is the number of SNPs in the proposed gene, though is certainly feasible for some small $n > 2$.

## Additional Graphics

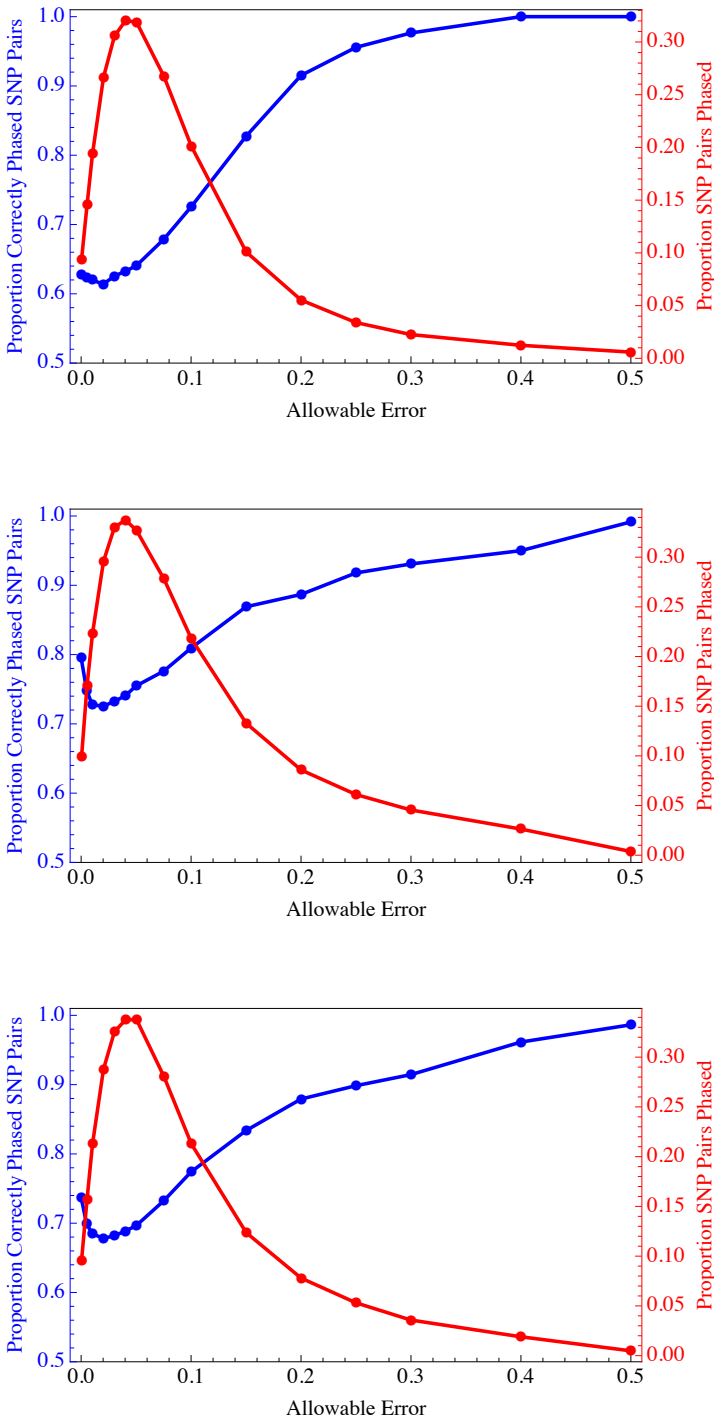


**Figure 6.13:** For varying allowed error, we report the proportion of SNP pairs with a unique feasible solution (red) and the accuracy of that feasible solution (AUFS) (blue) compared against the platinum phased VCF for (from top to bottom) Poly-A- replicate 1, Poly-A+ replicates 1 and 2.

## 6.7 Hyperplane Equations for $Z'(\hat{M}_4)$

For the interested reader, we have computed the equations defining the zonotope $Z'(\hat{M}_4)$. To get the equations for any zonotope defined by a subset of the isoforms of $\hat{M}_n$ for $n \leq 4$, set the relevant $\alpha_i = 0$. For example, setting $a_i = 0$ for $i > 3$ will give us the equations in 6.4.1. Below we show the particular representation of $\hat{M}_4$ used to define these equations. We present its transpose due to the length of the columns.

$$\text{Transpose}[M] = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix}$$

For this particular $\hat{M}_4$, the $A_j$ are the sums:
$A_1 = \alpha_1 + \alpha_3 + \alpha_5 + \alpha_7 + \alpha_9 + \alpha_{11} + \alpha_{13} + \alpha_{15}$
$A_2 = \alpha_2 + \alpha_3 + \alpha_6 + \alpha_7 + \alpha_{10} + \alpha_{11} + \alpha_{14} + \alpha_{15}$
$A_3 = \alpha_4 + \alpha_5 + \alpha_6 + \alpha_7 + \alpha_{12} + \alpha_{13} + \alpha_{14} + \alpha_{15}$
$A_4 = \alpha_8 + \alpha_9 + \alpha_{10} + \alpha_{11} + \alpha_{12} + \alpha_{13} + \alpha_{14} + \alpha_{15}$

The equations provided below have significant symmetry and we sort them by "Type" – plus-minus the multiset of coefficients attached to each $A_j x_j$. There are 45 in total; we do not have a formula for general $n$. For $n \in 1, 2, 3, 4$ they are $1, 3, 9, 45$.

Type: [1,0,0,0]
$|A_1 x_1| \leq \frac{\alpha_1+\alpha_3+\alpha_5+\alpha_7+\alpha_9+\alpha_{11}+\alpha_{13}+\alpha_{15}}{2}$
$|A_2 x_2| \leq \frac{\alpha_2+\alpha_3+\alpha_6+\alpha_7+\alpha_{10}+\alpha_{11}+\alpha_{14}+\alpha_{15}}{2}$
$|A_3 x_3| \leq \frac{\alpha_4+\alpha_5+\alpha_6+\alpha_7+\alpha_{12}+\alpha_{13}+\alpha_{14}+\alpha_{15}}{2}$
$|A_4 x_4| \leq \frac{\alpha_8+\alpha_9+\alpha_{10}+\alpha_{11}+\alpha_{12}+\alpha_{13}+\alpha_{14}+\alpha_{15}}{2}$

Type: [1,-1,0,0]
$|A_1 x_1 - A_2 x_2| \leq \frac{\alpha_1+\alpha_2+\alpha_5+\alpha_6+\alpha_9+\alpha_{10}+\alpha_{13}+\alpha_{14}}{2}$
$|A_1 x_1 - A_3 x_3| \leq \frac{\alpha_1+\alpha_3+\alpha_4+\alpha_6+\alpha_9+\alpha_{11}+\alpha_{12}+\alpha_{14}}{2}$
$|A_1 x_1 - A_4 x_4| \leq \frac{\alpha_1+\alpha_3+\alpha_5+\alpha_7+\alpha_8+\alpha_{10}+\alpha_{12}+\alpha_{14}}{2}$
$|A_2 x_2 - A_3 x_3| \leq \frac{\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_{10}+\alpha_{11}+\alpha_{12}+\alpha_{13}}{2}$
$|A_2 x_2 - A_4 x_4| \leq \frac{\alpha_2+\alpha_3+\alpha_6+\alpha_7+\alpha_8+\alpha_9+\alpha_{12}+\alpha_{13}}{2}$
$|A_3 x_3 - A_4 x_4| \leq \frac{\alpha_4+\alpha_5+\alpha_6+\alpha_7+\alpha_8+\alpha_9+\alpha_{10}+\alpha_{11}}{2}$

Type: [1,-1,-1,0]
$|A_1 x_1 - A_2 x_2 - A_3 x_3| \leq \frac{\alpha_1+\alpha_2+\alpha_4+2\alpha_6+\alpha_7+\alpha_9+\alpha_{10}+\alpha_{12}+2\alpha_{14}+\alpha_{15}}{2}$
$|A_1 x_1 - A_2 x_2 - A_4 x_4| \leq \frac{\alpha_1+\alpha_2+\alpha_5+\alpha_6+\alpha_8+2\alpha_{10}+\alpha_{11}+\alpha_{12}+2\alpha_{14}+\alpha_{15}}{2}$
$|A_1 x_1 - A_2 x_2 + A_3 x_3| \leq \frac{\alpha_1+\alpha_3+\alpha_4+\alpha_6+\alpha_8+\alpha_{10}+2\alpha_{12}+\alpha_{13}+2\alpha_{14}+\alpha_{15}}{2}$
$|A_1 x_1 - A_2 x_2 + A_4 x_4| \leq \frac{\alpha_1+\alpha_2+\alpha_4+2\alpha_5+\alpha_7+\alpha_9+\alpha_{10}+\alpha_{12}+2\alpha_{13}+\alpha_{15}}{2}$
$|A_1 x_1 + A_2 x_2 - A_3 x_3| \leq \frac{\alpha_1+\alpha_2+\alpha_5+\alpha_6+\alpha_8+2\alpha_9+\alpha_{11}+\alpha_{12}+2\alpha_{13}+\alpha_{15}}{2}$

$$|A_1x_1 - A_2x_2 + A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+\alpha_4+\alpha_7+\alpha_9+\alpha_{10}+2\alpha_{11}+\alpha_{12}+\alpha_{15}}{2}$$
$$|A_1x_1 + A_2x_2 - A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+\alpha_5+\alpha_6+2\alpha_7+\alpha_8+\alpha_{11}+\alpha_{12}+\alpha_{15}}{2}$$
$$|A_1x_1 + A_3x_3 - A_4x_4| \leq \frac{\alpha_1+\alpha_3+\alpha_4+2\alpha_5+\alpha_6+2\alpha_7+\alpha_8+\alpha_{10}+\alpha_{13}+\alpha_{15}}{2}$$
$$|A_1x_1 - A_3x_3 + A_4x_4| \leq \frac{\alpha_1+\alpha_3+\alpha_4+\alpha_6+\alpha_8+2\alpha_9+\alpha_{10}+2\alpha_{11}+\alpha_{13}+\alpha_{15}}{2}$$
$$|A_2x_2 - A_3x_3 - A_4x_4| \leq \frac{\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_8+\alpha_9+2\alpha_{12}+2\alpha_{13}+\alpha_{14}+\alpha_{15}}{2}$$
$$|A_2x_2 - A_3x_3 + A_4x_4| \leq \frac{\alpha_2+\alpha_3+\alpha_4+\alpha_5+\alpha_8+\alpha_9+2\alpha_{10}+2\alpha_{11}+\alpha_{14}+\alpha_{15}}{2}$$
$$|A_2x_2 + A_3x_3 - A_4x_4| \leq \frac{\alpha_2+\alpha_3+\alpha_4+\alpha_5+2\alpha_6+2\alpha_7+\alpha_8+\alpha_9+\alpha_{14}+\alpha_{15}}{2}$$

Type: [1,1,-1,-1]

$$|A_1x_1 + A_2x_2 - A_3x_3 - A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+\alpha_4+\alpha_7+\alpha_8+\alpha_{11}+2\alpha_{12}+\alpha_{13}+\alpha_{14}}{2}$$
$$|A_1x_1 - A_2x_2 + A_3x_3 - A_4x_4| \leq \frac{\alpha_1+\alpha_2+\alpha_4+2\alpha_5+\alpha_7+\alpha_8+2\alpha_{10}+\alpha_{11}+\alpha_{13}+\alpha_{14}}{2}$$
$$|A_1x_1 - A_2x_2 - A_3x_3 + A_4x_4| \leq \frac{\alpha_1+\alpha_2+\alpha_4+2\alpha_6+\alpha_7+\alpha_8+2\alpha_9+\alpha_{11}+\alpha_{13}+\alpha_{14}}{2}$$

Type: [1,1,1,-1]

$$|A_1x_1 - A_2x_2 - A_3x_3 - A_4x_4| \leq \frac{\alpha_1+\alpha_2+\alpha_4+2\alpha_6+\alpha_7+\alpha_8+2\alpha_{10}+\alpha_{11}+2\alpha_{12}+\alpha_{13}+3\alpha_{14}+2\alpha_{15}}{2}$$
$$|A_1x_1 - A_2x_2 + A_3x_3 + A_4x_4| \leq \frac{\alpha_1+\alpha_2+\alpha_4+2\alpha_5+\alpha_7+\alpha_8+2\alpha_9+\alpha_{11}+2\alpha_{12}+3\alpha_{13}+\alpha_{14}+2\alpha_{15}}{2}$$
$$|A_1x_1 + A_2x_2 - A_3x_3 + A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+\alpha_4+\alpha_7+\alpha_8+2\alpha_9+2\alpha_{10}+3\alpha_{11}+\alpha_{13}+\alpha_{14}+2\alpha_{15}}{2}$$
$$|A_1x_1 + A_2x_2 + A_3x_3 - A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+\alpha_4+2\alpha_5+2\alpha_6+3\alpha_7+\alpha_8+\alpha_{11}+\alpha_{13}+\alpha_{14}+2\alpha_{15}}{2}$$

Type: [1,1,1,-2]

$$|A_1x_1 + A_2x_2 + A_3x_3 - 2A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+\alpha_4+2\alpha_5+2\alpha_6+3\alpha_7+2\alpha_8+\alpha_9+\alpha_{10}+\alpha_{12}+\alpha_{15}}{2}$$
$$|A_1x_1 + A_2x_2 - 2A_3x_3 + A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+2\alpha_4+\alpha_5+\alpha_6+\alpha_8+2\alpha_9+2\alpha_{10}+3\alpha_{11}+\alpha_{12}+\alpha_{15}}{2}$$
$$|A_1x_1 - 2A_2x_2 + A_3x_3 + A_4x_4| \leq \frac{\alpha_1+2\alpha_2+\alpha_3+\alpha_4+2\alpha_5+\alpha_6+\alpha_8+2\alpha_9+\alpha_{10}+2\alpha_{12}+3\alpha_{13}+\alpha_{15}}{2}$$
$$|2A_1x_1 - A_2x_2 - A_3x_3 - A_4x_4| \leq \frac{2\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+2\alpha_6+\alpha_8+\alpha_9+2\alpha_{10}+2\alpha_{12}+3\alpha_{14}+\alpha_{15}}{2}$$

Type: [1,1,-1,-2]

$$|A_1x_1 - A_2x_2 - A_3x_3 + 2A_4x_4| \leq \frac{\alpha_1+\alpha_2+\alpha_4+2\alpha_6+\alpha_7+2\alpha_8+3\alpha_9+\alpha_{10}+2\alpha_{11}+\alpha_{12}+2\alpha_{13}+\alpha_{15}}{2}$$
$$|A_1x_1 - A_2x_2 + A_3x_3 - 2A_4x_4| \leq \frac{\alpha_1+\alpha_2+\alpha_4+2\alpha_5+\alpha_7+2\alpha_8+\alpha_9+3\alpha_{10}+2\alpha_{11}+\alpha_{12}+2\alpha_{14}+\alpha_{15}}{2}$$
$$|A_1x_1 + A_2x_2 - A_3x_3 - 2A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+\alpha_4+\alpha_7+2\alpha_8+\alpha_9+\alpha_{10}+3\alpha_{12}+2\alpha_{13}+2\alpha_{14}+\alpha_{15}}{2}$$
$$|A_1x_1 - A_2x_2 + 2A_3x_3 - A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_4+3\alpha_5+\alpha_6+2\alpha_7+\alpha_8+2\alpha_{10}+\alpha_{11}+\alpha_{12}+2\alpha_{13}+\alpha_{15}}{2}$$
$$|A_1x_1 - A_2x_2 - 2A_3x_3 + A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_4+\alpha_5+3\alpha_6+2\alpha_7+\alpha_8+2\alpha_9+\alpha_{11}+\alpha_{12}+2\alpha_{14}+\alpha_{15}}{2}$$
$$|A_1x_1 + A_2x_2 - 2A_3x_3 - A_4x_4| \leq \frac{\alpha_1+\alpha_2+2\alpha_3+2\alpha_4+\alpha_5+\alpha_6+\alpha_8+\alpha_{11}+3\alpha_{12}+2\alpha_{13}+2\alpha_{14}+\alpha_{15}}{2}$$
$$|A_1x_1 + 2A_2x_2 - A_3x_3 - A_4x_4| \leq \frac{\alpha_1+2\alpha_2+3\alpha_3+\alpha_4+\alpha_6+2\alpha_7+\alpha_8+\alpha_{10}+2\alpha_{11}+2\alpha_{12}+\alpha_{13}+\alpha_{15}}{2}$$
$$|A_1x_1 - 2A_2x_2 - A_3x_3 + A_4x_4| \leq \frac{\alpha_1+2\alpha_2+\alpha_3+\alpha_4+3\alpha_6+2\alpha_7+\alpha_8+2\alpha_9+\alpha_{10}+\alpha_{13}+2\alpha_{14}+\alpha_{15}}{2}$$
$$|A_1x_1 - 2A_2x_2 + A_3x_3 - A_4x_4| \leq \frac{\alpha_1+2\alpha_2+\alpha_3+\alpha_4+2\alpha_5+\alpha_6+\alpha_8+3\alpha_{10}+2\alpha_{11}+\alpha_{13}+2\alpha_{14}+\alpha_{15}}{2}$$
$$|2A_1x_1 + A_2x_2 - A_3x_3 - A_4x_4| \leq \frac{2\alpha_1+\alpha_2+3\alpha_3+\alpha_4+\alpha_5+2\alpha_7+\alpha_8+\alpha_9+2\alpha_{11}+2\alpha_{12}+\alpha_{14}+\alpha_{15}}{2}$$
$$|2A_1x_1 - A_2x_2 + A_3x_3 - A_4x_4| \leq \frac{2\alpha_1+\alpha_2+\alpha_3+\alpha_4+3\alpha_5+2\alpha_7+\alpha_8+\alpha_9+2\alpha_{10}+2\alpha_{13}+\alpha_{14}+\alpha_{15}}{2}$$
$$|2A_1x_1 - A_2x_2 - A_3x_3 + A_4x_4| \leq \frac{2\alpha_1+\alpha_2+\alpha_3+\alpha_4+\alpha_5+2\alpha_6+\alpha_8+3\alpha_9+2\alpha_{11}+2\alpha_{13}+\alpha_{14}+\alpha_{15}}{2}$$

# Bibliography


[1] URL: www.illumina.com/platinumgenomes/.

[2] GR Abecasis et al. "A map of human genome variation from population-scale sequencing." In: *Nature* 467.7319 (2010), pp. 1061–1073.

[3] Derek Aguiar and Sorin Istrail. "HapCompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data". In: *Journal of Computational Biology* 19.6 (2012), pp. 577–590.

[4] Derek Aguiar and Sorin Istrail. "Haplotype assembly in polyploid genomes and identical by descent shared tracts". In: *Bioinformatics* 29.13 (2013), pp. i352–i360.

[5] Vikas Bansal and Vineet Bafna. "HapCUT: an efficient and accurate algorithm for the haplotype assembly problem". In: *Bioinformatics* 24.16 (2008), pp. i153–i159.

[6] Vikas Bansal et al. "An MCMC algorithm for haplotype assembly from whole-genome sequence data". In: *Genome research* 18.8 (2008), pp. 1336–1346.

[7] Emily Berger et al. "HapTree: A Novel Bayesian Framework for Single Individual Polyplotyping Using NGS Data". In: *PLoS computational biology* 10.3 (2014), e1003502.

[8] Brian L Browning and Sharon R Browning. "A unified approach to genotype imputation and haplotype-phase inference for large data sets of trios and unrelated individuals". In: *The American Journal of Human Genetics* 84.2 (2009), pp. 210–223.

[9] Sharon R Browning and Brian L Browning. "High-resolution detection of identity by descent in unrelated individuals". In: *The American Journal of Human Genetics* 86.4 (2010), pp. 526–539.

[10] 1000 Genomes Project Consortium et al. "An integrated map of genetic variation from 1,092 human genomes". In: *Nature* 491.7422 (2012), pp. 56–65.

[11] Corrado De Concini and Claudio Procesi. *Topics in Hyperplane Arrangements, Polytopes and Box-Splines*. 0172-5939. Springer-Verlag New York, 2010.

[12] Olivier Delaneau, Cédric Coulonges, and Jean-François Zagury. "Shape-IT: new rapid and accurate algorithm for haplotype inference". In: *BMC bioinformatics* 9.1 (2008), p. 540.

[13] Alexander Dobin et al. "STAR: ultrafast universal RNA-seq aligner". In: *Bioinformatics* 29.1 (2013), pp. 15–21.

[14] Anatoly Efros and Eran Halperin. "Haplotype reconstruction using perfect phylogeny and sequence data". In: *BMC bioinformatics* 13.Suppl 6 (2012), S3.

[15] Uriel Feige. "Coping with the NP-Hardness of the Graph Bandwidth Problem". English. In: *Algorithm Theory - SWAT 2000*. Vol. 1851. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2000, pp. 10–19. ISBN: 978-3-540-67690-4. DOI: 10.1007/3-540-44985-X_2. URL: http://dx.doi.org/10.1007/3-540-44985-X_2.

[16] Filippo Geraci. "A comparison of several algorithms for the single individual SNP haplotyping reconstruction problem". In: *Bioinformatics* 26.18 (2010), pp. 2217–2225.

[17] Jennifer Harrow et al. "GENCODE: the reference human genome annotation for The ENCODE Project". In: *Genome research* 22.9 (2012), pp. 1760–1774.

[18] Dan He et al. "Optimal algorithms for haplotype assembly from whole-genome sequence data". In: *Bioinformatics* 26.12 (2010), pp. i183–i190.

[19] John Hopcroft and Robert Tarjan. "Algorithm 447: Efficient Algorithms for Graph Manipulation". In: *Commun. ACM* 16.6 (June 1973), pp. 372–378. ISSN: 0001-0782. DOI: 10.1145/362248.362272. URL: http://doi.acm.org/10.1145/362248.362272.

[20] Giuseppe Lancia et al. "SNPs problems, complexity, and algorithms". In: *Algorithms—ESA 2001*. Springer, 2001, pp. 182–193.

[21] Heng Li and Richard Durbin. "Fast and accurate short read alignment with Burrows–Wheeler transform". In: *Bioinformatics* 25.14 (2009), pp. 1754–1760.

[22] Ross Lippert et al. "Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem". In: *Briefings in bioinformatics* 3.1 (2002), pp. 23–31.

[23] L. Pachter. "Models for transcript quantification from RNA-Seq". In: (2011). URL: arXiv:1104.3889v2.

[24] Alessandro Panconesi and Mauro Sozio. "Fast hare: A fast heuristic for single individual SNP haplotype reconstruction". In: *Algorithms in Bioinformatics*. Springer, 2004, pp. 266–277.

[25] Andrew Quinn, Punita Juneja, and Francis M Jiggins. "Estimates of allele-specific expression in Drosophila with a single genome sequence and RNA-seq data". In: *Bioinformatics* (2014), btu342.

[26] Adam Roberts. "Ambiguous fragment assignment for high-throughput sequencing experiments". PhD thesis. EECS Department, University of California, Berkeley, Oct. 2013. URL: http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-177.html.

[27] Adam Roberts and Lior Pachter. "Streaming fragment assignment for real-time analysis of sequencing experiments". In: *Nat Meth* 10.1 (Jan. 2013), pp. 71–73. URL: http://dx.doi.org/10.1038/nmeth.2251.

[28] Kate R Rosenbloom et al. "ENCODE data in the UCSC Genome Browser: year 5 update". In: *Nucleic acids research* 41.D1 (2013), pp. D56–D63.

[29] Paul Scheet and Matthew Stephens. "A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase". In: *The American Journal of Human Genetics* 78.4 (2006), pp. 629–644.

[30] Matthew Stephens, Nicholas J Smith, and Peter Donnelly. "A new statistical method for haplotype reconstruction from population data". In: *The American Journal of Human Genetics* 68.4 (2001), pp. 978–989.

[31] Amy L Williams et al. "Rapid haplotype inference for nuclear families". In: *Genome biology* 11.10 (2010), R108.

[32] Kui Zhang, Fengzhu Sun, and Hongyu Zhao. "HAPLORE: a program for haplotype reconstruction in general pedigrees without recombination". In: *Bioinformatics* 21.1 (2005), pp. 90–103.