The Dissertation Committee for Soyeon Ahn
certifies that this is the approved version of the following dissertation:

Computational Methods for Understanding Genetic Variations from Next Generation Sequencing Data

Committee:

Haris Vikalo, Supervisor

Gustavo de Veciana

Sriram Vishwanath

David Soloveichik

Cagri Savran

Computational Methods for Understanding Genetic Variations from Next Generation Sequencing Data

by

Soyeon Ahn,

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2018

Dedicated to family for their endless encouragement and belief in me

Acknowledgments

First of all, I would like to express my gratitude to advisor, Prof. Haris Viaklo, for his continuous encouragement, advice, endless patience and generosity throughout the course of my PhD. As well as giving me a lot of academic skills, he has constantly guided me on the right path to develop and fill my weaknesses and tried to raise my potential. Being a graduate student of him is one of the greatest luck in my life. Without him, this dissertation could not have been done.

I would like to thank the members of my dissertation committees, Prof. Gustavo de Veciana, Prof. Sriram Vishwanath, Prof. David Soloveichik and Prof. Cagri Savran for their valuable time, insightful comments and thought-provoking questions that broadened my perspective of thinking on research topics and problems. In addition, I also thank the ECE administrative staffs, Melanie Gullick, Andrew Kieschnick, Melody Singleton, Apipol Piman and Karen Little for always making them available and giving me help whenever I needed on various issues.

To my excellent collaborators and lab mates, Ziqi ke, Somsubhra Barik, Natalia Arzeno-Gonzlez, Abolfazl Hashemi, Shreepriya Das, Xiaohu Shen, thank you for stimulating discussions on research and contributing to my learning. To my friends in WNCG and AKPC, thank you for being with me and

enriching my life at Austin.

Last but not least, I give the greatest gratitudes to my family. Thank Mom and Dad for your unconditional love, consistent support and encouragement, and belief in me. Thank Sunyoung for being my sister. I cannot wait to become an aunt. Finally, to my husband, Wonkyum, I cannot express my gratitude enough for putting up with me, sharing all the joy and frustration that I've experienced during my PhD and being my biggest supporter. For all this work, give glory to God.

Computational Methods for Understanding Genetic Variations from Next Generation Sequencing Data

Publication No. _____

Soyeon Ahn, Ph.D.
The University of Texas at Austin, 2018

Supervisor: Haris Vikalo

Studies of human genetic variation reveal critical information about genetic and complex diseases such as cancer, diabetes and heart disease, ultimately leading towards improvements in health and quality of life. Moreover, understanding genetic variations in viral population is of utmost importance to virologists and helps in search for vaccines. Next-generation sequencing technology is capable of acquiring massive amounts of data that can provide insight into the structure of diverse sets of genomic sequences. However, reconstructing heterogeneous sequences is computationally challenging due to the large dimension of the problem and limitations of the sequencing technology.

This dissertation is focused on algorithms and analysis for two problems in which we seek to characterize genetic variations: (1) haplotype reconstruction for a single individual, so-called *single individual haplotyping* (SIH) or *haplotype assembly* problem, and (2) reconstruction of viral population, the

so-called *quasispecies reconstruction* (QSR) problem. For the SIH problem, we have developed a method that relies on a probabilistic model of the data and employs the sequential Monte Carlo (SMC) algorithm to jointly determine type of variation (i.e., perform genotype calling) and assemble haplotypes. For the QSR problem, we have developed two algorithms. The first algorithm combines agglomerative hierarchical clustering and Bayesian inference to reconstruct quasispecies characterized by low diversity. The second algorithm utilizes tensor factorization framework with successive data removal to reconstruct quasispecies characterized by highly uneven frequencies of its components. Both algorithms outperform existing methods in both benchmarking tests and real data.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Genetic variations predispose individuals to hereditary diseases, play important role in the development of complex diseases, and impact drug metabolism. In diploid organisms, genomes are organized into pairs of chromosomes, one inherited from each of the parents. Single Nucleotide Polymorphism (SNP), the most common form of genetic variations, refers to an event where the bases (A,C,G,T) in a specific position of the chromosomes in a pair differ from each other. For humans, SNPs occur on average once in 300 bases [45]. The position where the chromosomes in a pair have the same base is called a homozygous site; otherwise, it is called a heterozygous site. The full information about the DNA variations in the genome of an individual is given by haplotypes, the ordered lists of single nucleotide polymorphisms (SNPs) located on chromosomes. Haplotype information has been of great importance for studies of human diseases and effectiveness of drugs [33], which motivated considerable efforts towards developing methods for haplotype identification and analysis.

In recent years, inferring haplotype information has been enabled by next-generation sequencing technologies, where a genome is sampled by a large

collection of pairs of short reads whose relative positions are determined by mapping to a reference genome. The goal of haplotype assembly is to reconstruct haplotype sequences from the collection of reads randomly sampled from two chromosomes of an individual genome. However, haplotype assembly is a challenging problem due to the limited information about the SNP sites provided by short reads and inherent randomness of the sequencing process.

While the information of haplotype, i.e., an ordered collection of DNA sequence variations, is invaluable for studies of the genetics of common diseases, a number of potentially life-threatening infectious diseases are caused by RNA viruses, including human immunodeficiency virus (HIV), hepatitis C virus (HCV), influenza, Ebola and Zika. The RNA viruses are characterized by high mutation rates that lead to new viral strains by means of point mutations, insertions and deletions. The resulting population of closely related yet non-identical viral genomes is known as a viral quasispecies [41]. Genetic heterogeneity of such viral populations enables the virus to adapt and proliferate in dynamically changing environments, e.g. over the course of an infection [15]. Therefore, determining the genetic structure of viral quasispecies is of importance for effective antiviral vaccine designs and the development of therapeutic treatments for viral diseases.

Quasispecies Spectrum Reconstruction (QSR) aims to reconstruct an a priori unknown number of viral sequences in a quasispecies and estimate their relative frequencies. To this end, QSR methods typically employ the following steps: (1) clustering together sequencing reads that originate from the

same strain; (2) reconstructing the strains using the clustered reads; and (3) determining relative frequencies of the reconstructed strains based on the corresponding cluster sizes [13]. High-throughput sequencing (HTS) technologies have in principle enabled unprecedented studies of quasispecies populations. However, their precise reconstruction remains difficult due to the presence of sequencing errors and limited length of HTS reads. The QSR problem is particularly challenging when the frequencies of strains in a viral population are highly imbalanced, i.e., the quasispecies contains both strains having high and those having low abundances, and is further exacerbated if the genetic distances between strains are relatively small. In those settings, performance of clustering-based QSR methods suffers from erroneous attribution of the reads that have originated from rare strains to nearby (in terms of genetic distance) highly abundant strains; such errors lead to failures to discover strains of low abundancy [56] and thus may hinder the discovery of effective drug treatments [69, 42].

Driven by the importance of precise characterization of genetic variations in medical genomics and its potentials for improving human health, the goal of this dissertation is to study two problems: haplotype assembly and quasispecies reconstruction. We focus on developing statistical and computational algorithms that take into account the intrinsic features of those problems.

1.1 Motivation and Contributions

A. Haplotype assembly algorithm and analysis

Affordable high-throughput DNA sequencing technologies enable routine acquisition of data needed for the assembly of single individual haplotypes. However, state-of-the-art high-throughput sequencing platforms generate data that is erroneous, which induces uncertainty in the SNP and genotype calling procedures and, ultimately, adversely affect the accuracy of haplotyping. When inferring haplotype phase information, the vast majority of the existing techniques for haplotype assembly assume that the genotype information is correct. This motivates the development of methods capable of joint genotype calling and haplotype assembly.

We present a haplotype assembly algorithm, ParticleHap [4], that relies on a probabilistic description of the sequencing data to jointly infer genotypes and assemble the most likely haplotypes. Our method employs a deterministic sequential Monte Carlo algorithm that associates single nucleotide polymorphisms with haplotypes by exhaustively exploring all possible extensions of the partial haplotypes. The algorithm relies on genotype likelihoods rather than on often erroneously called genotypes, thus ensuring a more accurate assembly of the haplotypes. The achievable performance of the algorithm is analyzed. Results on both the 1000 Genomes Project experimental data as well as simulation studies demonstrate that the proposed approach enables highly accurate solutions to the haplotype assembly problem while being computationally efficient and scalable, generally outperforming existing methods in terms of both accuracy and speed.

B. Algorithms for quasispecies reconstruction

RNA viruses replicate with high mutation rates, creating closely related viral populations. The heterogeneous virus populations, referred to as viral quasispecies, rapidly adapt to environmental changes thus adversely affecting efficiency of antiviral drugs and vaccines. Therefore, studying the underlying genetic heterogeneity of viral populations plays a significant role in the development of effective therapeutic treatments. Recent high-throughput sequencing technologies have provided invaluable opportunity for uncovering the structure of quasispecies populations. However, sequencing errors and limited read lengths render the problem of reconstructing the strains and estimating their spectrum challenging. The problem is particularly challenging when the strains in a population are highly similar, i.e., the sequences are characterized by low mutual genetic distances, and further exacerbated if some of those strains are relatively rare due to generally non-uniform frequencies of the strains; this is the setting where state-of-the-art methods struggle.

Motivated by those challenges, we present two algorithms that focus on reconstruction of viral population characterized by low diversity and imbalanced frequencies of strains. The first viral quasispecies reconstruction algorithm, aBayesQR [5], employs a maximum-likelihood framework to infer individual sequences in a mixture from high-throughput sequencing data. The search for the most likely quasispecies is conducted on long contigs that our method constructs from the set of short reads via agglomerative hierarchical clustering; operating on contigs rather than short reads enables identification of close strains in a population and provides computational tractability of the

Bayesian method. Results on both simulated and real HIV-1 data demonstrate that the proposed algorithm generally outperforms state-of-the-art methods; aBayesQR particularly stands out when reconstructing a set of closely related viral strains (e.g., quasispecies characterized by low diversity).

The second algorithm, TenSQR, utilizes tensor factorization framework to analyze high-throughput sequencing data and reconstruct viral quasispecies characterized by highly uneven frequencies of its components. Fundamentally, TenSQR performs clustering with successive data removal to infer strains in a quasispecies in order from the most to the least abundant one; every time a strain is inferred, sequencing reads generated from that strain are removed from the dataset. The proposed successive strain reconstruction and data removal enables discovery of rare strains in a population and facilitates detection of deletions in such strains. Results on simulated datasets demonstrate that TenSQR can reconstruct full-length strains having widely different abundances, generally outperforming state-of-the-art methods at diversities 1-10% and detecting long deletions even in rare strains. A study on a real HIV-1 dataset demonstrates that TenSQR outperforms competing methods in experimental settings as well. Finally, we apply TenSQR to analyze a Zika virus sample and reconstruct the full-length strains it contains.

1.2 Organization

The rest of the dissertation is organized as follows. Chapter 2 presents an algorithm for haplotype assembly and analysis of achievable performance.

Chapter 3 presents an algorithm to reconstruct viral quasispecies characterized by low diversity using agglomerative hierarchical clustering and Bayesian inference. Chapter 4 presents another algorithm for reconstruction of viral quasispecies when the frequencies of strains in a quasispecies are highly imbalanced. Chapter 5 summarizes the ideas of this dissertation and suggest future research directions fueled by the work presented in previous chapters.

Chapter 2

Haplotype assembly and its Analysis

2.1 Background and Related Works

Increased affordability of high-throughput DNA sequencing has enabled studies of genetic variations and of the effects they have on health and medical treatments. In diploid organisms, such as humans, chromosomes come in pairs. The chromosomes in a pair of autosomes are homologous, i.e., they have similar composition and carry the same type of information but are not identical. The most common type of DNA sequence variation is a single nucleotide polymorphism (SNP), where a single base in the genome differs between individuals or paired chromosomes. Each of those variants is referred to as an allele; a SNP has at least two different alleles. If the two alleles at a SNP site are same, the SNP site is homozygous; if they are different, it is heterozygous. SNP calling is concerned with identification of the locations and types of such alleles, and is followed by genotype calling to decide the genotypes associated with the locations of the detected SNPs. Accurate SNP and genotype calling are challenging due to uncertainties caused by base calling and read alignment errors. The low-to-medium coverages typical of large-scale sequencing projects are often associated with erroneous SNP and genotype calling [54]. As an illustration, in the low-coverage ($2-6\times$) 1000 Genomes Project pilot, the geno-

type accuracy at heterozygous sites was 90% for the lowest allele frequencies (minor allele frequency (MAF)$<$ 3%), 95% for the intermediate frequencies (MAF 50%), and $70-80\%$ for the highest frequency variants (MAF$>$ 97%) [20].

SNP and genotype calling do not assign alleles to specific chromosomes in the pairs. Such detailed information is provided by haplotypes, ordered collections of SNPs on the chromosomes. Haplotypes have been of fundamental importance for the studies of human diseases and effectiveness of drugs [33]. The International Haplotype Map Project's pursuit of developing a haplotype map of the human genome reflects the significance of acquiring and understanding haplotype information [21]. *Haplotype inference* typically refers to the task of reconstructing haplotypes from the genotype samples of a population. *Haplotype assembly*, or single individual haplotyping, aims to reconstruct single individual haplotypes from high-throughput sequencing data. Since the SNP sites are assumed to be bi-allelic (i.e., each SNP site contains one of only two possible nucleotides), the alleles are labelled as 0 and 1 and the haplotypes are represented by binary sequences. Therefore, haplotype assembly is often cast as the problem of phasing two binary sequences from their short samples (i.e., reads) that are represented by ternary strings (where the third symbol denotes missing information). Majority of the existing haplotype assembly algorithms rely on this formulation of the problem [67].

Several haplotype assembly criteria and algorithms to optimize them were considered in [38, 48]. The minimum error correction (MEC) criterion,

in particular, has received a considerable amount of attention and has been broadly used in practice. Most of the haplotype assembly problem formulations have been shown to be NP-hard [38, 48, 19], which has motivated numerous computationally efficient heuristic solutions [67]. FastHare, proposed in [1], was an early heuristic method that was followed by several approximate techniques in [43, 17]. In [81, 74], the use of clustering approaches for splitting reads into two sets, each associated with one chromosome in a pair, was proposed. In addition to the approximate methods, several algorithms that search for the exact solution to the MEC formulation of the problem were developed, including the branch-and-bound technique in [73]. However, as argued in [30], the exact algorithms are often infeasible in practice; the approach in [10] based on the Markov Chain Monte Carlo (MCMC) method, HASH, also incurs high computational cost while being more accurate than heuristics. As a follow-up to HASH, [9] presented a significantly faster heuristic algorithm, HapCUT, suffering only a minor loss of accuracy. To minimize the MEC score, HapCUT iteratively computes the max-cut in a graph that represents the assembly problem. In [26], another max-cut based heuristic, ReFHap, was proposed; ReFHap relies on a different graph structure to achieve higher speed while maintaining accuracy similar to that of HapCUT. Other methods include a dynamic programming solution in [32]; a method that solves an appropriate integer linear program [18]; and several other heuristics including [22], H-BOP [77], HMEC [12], and HapCompass [2, 3].

A probabilistic framework for haplotype assembly was first introduced

in [45]. There, in order to deal with inherently random errors in sequencing data, the probability that a site in the fragments is incorrectly sequenced is defined for each of the four nucleotide bases. The most likely haplotype phases between SNP sites are determined using joint posteriori probabilities whose calculation is limited to two or three adjacent SNPs due to the intensive computational cost that grows exponentially in the number of SNP sites. The locally estimated haplotype segments are linked if the corresponding confidence levels exceed a certain threshold. In the follow-up work [75], reconstruction of longer haplotype segments using the Gibbs sampling procedure was enabled. However, this iterative approach still first assembles short haplotype segments that are then connected, and requires runtimes infeasible for block lengths typically encountered in practice. More recently, [52] proposed a new probabilistic mixture model, MixSIH, which leads to a more efficient computation of the haplotype likelihoods than those in [45, 75]. However, MixSIH is still about 10-fold slower than either HapCUT [9] or ReFHap [26] while having comparable accuracy, and the model there is restricted to the bi-allelic representation.

Increasing the number of reads covering each SNP position, so-called sequencing coverage, enables improving the accuracy of haplotype assembly. However, this also increases the sequencing cost and decreases the speed of haplotype assembly. Recently, [68] studied the problem from an information-theoretic perspective, presenting lower bounds of the number of reads required for near-perfect haplotype assembly. Furthermore, [37] derived optimal bounds

for near-perfect reconstruction by connecting the haplotype assembly problem with that of decoding convolutional codes.

It is worth pointing out that most of the existing algorithms for haplotype assembly allow no more than two alleles at a SNP site and only deal with the errors caused by substitutions between those two alleles. In practice, when sequencing errors lead to reads that report more than two alleles at a SNP site, either all of the tri- or tetra-allelic sites are discarded [43, 9] or the alleles that do not match the reference (or its alternative) are thrown away [2, 3]. The former drastically reduces not only the number of SNP sites to be reconstructed but also the chance of reliable full haplotype reconstruction (due to reducing the length of already short reads). The main drawback to the latter is that by fully trusting genotype information provided by SNP/genotype calling, the true haplotypes may be incorrectly reconstructed when the genotype calling is erroneous (i.e., when alleles corresponding to an incorrect genotype are preserved while the alleles corresponding to the true genotype are discarded).

In this chapter, we present a novel method that relies on a probabilistic model of the data to incorporate genotype calling in the haplotype assembly procedure [4][1]. Unlike [45, 75], the proposed method infers both the most likely genotypes and haplotype phases by examining the complete set of SNP loci in a computationally efficient manner. To this end, we employ the se-

[1]This work has been published as [Soyeon Ahn and Haris Vikalo. Joint haplotype assembly and genotype calling via sequential monte carlo algorithm. BMC bioinformatics, 16(1):223, 2015]. The author of this dissertation is the primary contributor.

quential Monte Carlo (SMC) algorithm (i.e., a particle filter). Particle filters are capable of sequentially estimating the posterior density of unknown variables by representing them with a set of particles and associated weights [7]. When the solution space is discrete and finite, a deterministic form of SMC can be derived [29, 61]; this has been exploited for solving various problems in genomics [46, 47]. Noting that the set of possible haplotype pairs is discrete and finite, we develop a modified deterministic sequential Monte Carlo (DSMC) method for solving the haplotype assembly problem. Our algorithm, ParticleHap, relies on the 2^{nd}-order Markov model of the haploype sequence to search for the most likely association of the SNPs to haplotypes. Phasing of the SNPs is done sequentially: the posteriori probability of a partial haplotype comprising n SNPs is calculated using the read information about the SNP in the n^{th} position of the haplotype sequence and the posteriori probability of the previously inferred $(n-1)$-bases long partial haplotype. By working with SNP calls rather than their binary representations (the latter is typically used by state-of-the-art haplotype assembly algorithms), ParticleHap can reliably infer the most likely genotypes. The performance analysis of ParticleHap is followed. Our extensive computational studies demonstrate that the proposed scheme is more accurate and computationally more efficient than state-of-the-art methods in [9, 26].

2.2 ParticleHap: Joint haplotype assembly and genotype calling algorithm

2.2.1 Problem formulation

In our model and the subsequently proposed haplotype assembly algorithm, we focus on the SNP sites where two or more alleles are observed. The sites with only one observed allele are declared homozygous and not used for the assembly. Assume there are m paired-end short reads covering n remaining SNP sites. Such data can be represented by an $m \times n$ matrix where the rows contain information provided by the reads while the columns correspond to the SNP sites.

By adopting the notation used in [45, 75], let $\mathbf{X}$ with elements $X_{ij} = x_{ij}, x_{ij} \in \mathcal{B}$, be the matrix of potentially erroneous observations, while $\mathbf{Y}$ with entries $Y_{ij} = y_{ij}, y_{ij} \in \mathcal{A}$, denote the corresponding error-free data matrix, $1 \leq i \leq m$, $1 \leq j \leq n$, $\mathcal{A} = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ and $\mathcal{B} = \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}, -\}$. Here $-$ denotes a gap, i.e., a site not covered by a read or an ambiguous base-call. Let a $2 \times n$ matrix $\mathbf{S}$ with entries $S_{kj} = s_{kj}$, $s_{kj} \in \mathcal{A}$, $k \in \{1, 2\}$, denote the true haplotype pair, and let us collect the indicators of the origin of the reads, $f_i \in \{0, 1\}$, $1 \leq i \leq m$, into a vector $\mathbf{F}$. With this notation, the true bases relate to the true haplotypes as $Y_{ij} = S_{f_i,j}$, where $1 \leq i \leq m$, $1 \leq j \leq n$. We assume that the composition probabilities $\Pr(S_{kj} = s)$, $s \in \mathcal{A}$, at each SNP position are mutually independent and constant across haplotypes. The measurement model is given by $\Pr(X_{ij} = x_{ij} | Y_{ij} = y_{ij})$, $y_{ij} \in \mathcal{A}$, $x_{ij} \in \mathcal{B}$. A sequencing error occurs when the true base $Y_{ij} = y_{ij}$ is misread as $X_{ij} = x_{ij}$, $x_{ij} \neq y_{ij}$. Note that the posteriori probability $p(\mathbf{S}|\mathbf{X})$ can be computed from

$p(\mathbf{S}, \mathbf{X})$ and $p(\mathbf{X}) = \sum_{\mathbf{S}} p(\mathbf{S}, \mathbf{X})$ using the Bayes' rule, where

$$\begin{aligned} p(\mathbf{S}, \mathbf{X}) &= \left(\frac{1}{2}\right)^m \prod_{j=1}^{n} p(S_{1j}) \prod_{j=1}^{n} p(S_{2j}) \\ &\times \prod_{i=1}^{m} \left[\prod_{j=1}^{n} p(X_{ij} \mid S_{1j}) + \prod_{j=1}^{n} p(X_{ij} | S_{2j}) \right]. \end{aligned}$$

We assume that each read is generated from one of the two haplotypes with probability $\frac{1}{2}$, i.e., $\Pr(f_i = 0) = \Pr(f_i = 1) = \frac{1}{2}$.

2.2.2 ParticleHap algorithm

Following the adopted notation, the goal of haplotype assembly is to determine matrix $\mathbf{S}$ from the observation matrix $\mathbf{X}$. A Bayesian approach to solving this problem involves maximization of the posteriori distribution $p(\mathbf{S}|\mathbf{X})$. Let $S_{\cdot j}$ and $X_{\cdot j}$ denote the j^{th} column vectors of $\mathbf{S}$ and $\mathbf{X}$, respectively, and let us define $\mathrm{S}_{1:t} = \{S_{\cdot 1}, S_{\cdot 2}, \cdots, S_{\cdot t}\}$ and $\mathrm{X}_{1:t} = \{X_{\cdot 1}, X_{\cdot 2}, \cdots, X_{\cdot t}\}$. Recursive Bayesian estimation (i.e., Bayesian filtering) is concerned with recursively finding the conditional probability density function $p(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})$. Having obtained the estimate $\hat{p}(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})$, we can determine the most likely $\mathrm{S}_{1:t}$. However, finding an analytical form of this probability density function is often infeasible, as is the case for the haplotype assembly problem.

Sequential Monte Carlo (SMC), often referred to as particle filtering [7], describes $p(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})$ using a set of discrete points (particles) and their corresponding weights. SMC can be interpreted as the dynamical system which, in the context of haplotype assembly, comprises the initial state model

$p(S_{\cdot 1})$, state transitions model $p(S_{\cdot t}|S_{\cdot t-1})$ and measurement model $p(X_{\cdot t}|S_{\cdot t})$. The distribution $p(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})$ can be propagated using an importance sampling technique where samples from a proposal density $q(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})$ are generated and appropriately weighted. Having drawn K samples $\{S_{\cdot t}^{(1)}, S_{\cdot t}^{(2)}, \cdots, S_{\cdot t}^{(K)}\}$ from $q(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})$ and assigned them weights $w_t^{(k)}$, $p(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})$ can be approximated by

$$\hat{p}(\mathrm{S}_{1:t}|\mathrm{X}_{1:t}) = \frac{1}{W_t}\sum_{k=1}^{K} w_t^{(k)}\delta(\mathrm{S}_{1:t} - \mathrm{S}_{1:t}^{(k)}),\ w_t^{(k)} = \frac{p(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})}{q(\mathrm{S}_{1:t}|\mathrm{X}_{1:t})}, \tag{2.1}$$

where $W_t = \sum_{k=1}^{K} w_t^{(k)}$ and $\delta(\cdot)$ is an indicator function, i.e., $\delta(s - s_0) = 1$ for $s = s_0$ and $\delta(s - s_0) = 0$ otherwise. The weight $w_t^{(k)}$ can be further derived as [7]

$$\begin{aligned} w_t^{(k)} &\propto w_{t-1}^{(k)} p(X_{\cdot t} \mid S_{\cdot t-1}^{(k)}) \\ &\propto w_{t-1}^{(k)} \sum_{l=1}^{L} p(X_{\cdot t} \mid S_{\cdot t} = s_l) p(S_{\cdot t} = s_l \mid S_{\cdot t-1}^{(k)}). \end{aligned}$$

In the traditional SMC, the set $\{(\mathrm{S}_{1:t}^{(k)}, w_t^{(k)}), k = 1, \cdots, K\}$ is recursively generated from the previous set of properly weighted samples $\{(\mathrm{S}_{1:t-1}^{(k)}, w_{t-1}^{(k)}), k = 1, \cdots, K\}$ by using the optimal proposal distribution $q(S_{\cdot t}|\mathrm{S}_{1:t-1}^{(k)}, \mathrm{X}_{1:t}) = p(S_{\cdot t}|\mathrm{S}_{1:t-1}^{(k)}, \mathrm{X}_{1:t})$,

$$q(S_{\cdot t} = s_l|\mathrm{S}_{1:t-1}^{(k)}, \mathrm{X}_{1:t}) \propto p(X_{\cdot t}|S_{\cdot t} = s_l) p(S_{\cdot t} = s_l|S_{\cdot t-1}^{(k)}).$$

In contrast to the conventional SMC, the *deterministic sequential Monte Carlo* (DSMC, [29, 61]) explores all possible states in each step of the recursive procedure. In particular, each particle at step $t-1$, $S_{\cdot t-1}^{(k)}, k = 1, \cdots, K$,

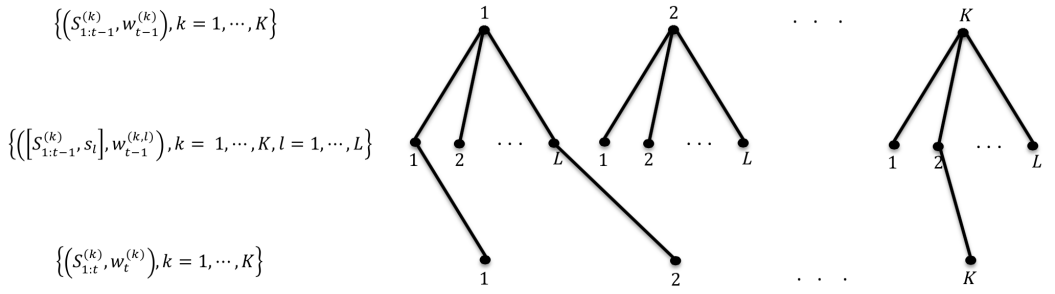

Figure 2.1: Procedure of propagating particles in deterministic sequential Monte Carlo. Each of K particles at step $t-1$ is propagated to L possible states at step t. Among $K \times L$ possible particles, only K particles with the highest weight are selected.

is propagated to L possible states at step t instead of being propagated to a single particle, where L denotes the number of possible extensions of the partially reconstructed haplotype sequence in our haplotype assembly problem. Maintaining and further propagating all such particles would inevitably increase their number exponentially; to remedy this problem, in each step only K particles with the highest weights among KL possible particles are selected. Then, given a set $\{(\mathrm{S}_{1:t-1}^{(k)}, w_{t-1}^{(k)}), k = 1, \ldots, K\}$ that does not contain duplicate paths, (2.1) and Bayes' theorem lead to an approximation of the posterior distribution of $\mathrm{S}_{1:t}$

$$\hat{p}^{\mathrm{DSMC}}(\mathrm{S}_{1:t}|\mathrm{X}_{1:t}) = \frac{1}{W_t^{\mathrm{DSMC}}} \sum_{k=1}^{K} \sum_{l=1}^{L} w_t^{(k,l)} \delta(\mathrm{S}_{1:t} - [\mathrm{S}_{1:t-1}^{(k)}\ s_l]),$$

where $W_t^{\mathrm{DSMC}} = \sum_{k,i} w_t^{(k,l)}$ and $[\mathrm{S}_{1:t-1}^{(k)}, s_l]$ is obtained by appending the state s_l to $\mathrm{S}_{1:t-1}^{(k)}$. Each weight $w_t^{(k,l)}$ is calculated as

$$w_t^{(k,l)} \propto w_{t-1}^{(k)} p(X_{\cdot t}|S_{\cdot t} = s_l) p(S_{\cdot t} = s_l|S_{\cdot t-1}^{(k)}). \tag{2.2}$$

The procedure is continued until obtaining $\mathrm{S}_{1:n}^{(k)} = (\mathrm{S}_{1:n-1}^{(k)}, S_{\cdot n}^{(k)})$ and its corresponding weights. Figure 2.1 illustrates the procedure of propagating particles in the DSMC. Each of K particles at step $t-1$ is propagated to L possible states at step t. Among $K \times L$ possible particles, only K particles with the highest weight are selected.

The conditional distribution $p(X_{\cdot t}|S_{\cdot t} = s_l)$ in the DSMC weight updates (2.2) reflects dependence of $X_{\cdot t}$ on the current state $S_{\cdot t}$ only, and does not include the phase information between nearby SNP sites (note that it does enable detection of the most likely genotypes at the t^{th} site). To incorporate the phasing information, we extend the representation of the particle trajectories to the 2^{nd}-order Markov model. In particular, we modify (2.2) so that the weight updates in our ParticleHap depend on the history of the state and the observation at $t-1$ as well as on the current state,

$$w_t^{(k,l)} \propto w_{t-1}^{(k)} p(X_{\cdot t}|S_{\cdot t} = s_l, S_{\cdot t-1}^{(k)}, X_{\cdot t-1}) p(S_{\cdot t} = s_l|S_{\cdot t-1}^{(k)}, X_{\cdot t-1}), \tag{2.3}$$

where $s_l = (s_{l_1}, s_{l_2})$, $l = 1, \ldots, L$. In particular, at step t, ParticleHap examines potential extensions of the partially reconstructed haplotype by adding a single SNP site, which requires no more than 12 likelihood calculations – one for each possible heterozygous pair (s_{l_1}, s_{l_2}) at site t, with 12 such pairs when there are 4 different bases in the t^{th} column of $\mathbf{X}$.

While the conditioning on $S_{\cdot t-1}^{(k)}$ and $X_{\cdot t-1}$ in $p(X_{\cdot t}|S_{\cdot t} = s_l, S_{\cdot t-1}^{(k)}, X_{\cdot t-1})$ in (2.3) introduced phase information between SNPs in positions $t-1$ and t, there remains a major challenge for reconstruction of unknown haplotype

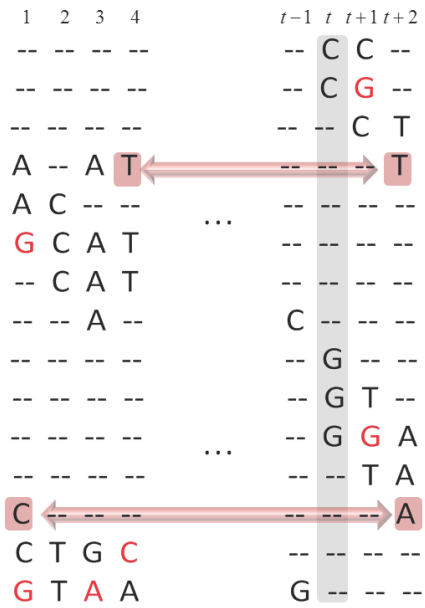

Figure 2.2: Information about heterozygous sites provided by paired-end reads and organized in the observation matrix $\mathbf{X}$. Erroneous base characters are highlighted in red font.

due to gaps in the data matrix $\mathbf{X}$. Even with the previously described 2^{nd}-order Markov model of particle trajectories, phase information between two consecutive SNP sites cannot be retrieved using a read that is not covering both of those sites. For example, in Figure 2.2, column $t+2$ contains 5 informative entries (i.e., entries which are not $-$). However, 2 of them belong to the reads which do not cover the SNP in the position $t+1$, i.e., are immediately preceded by $-$, and thus do not contribute to the phase information if the weights are computed according to (2.3). To this end, we modify (2.3) so that the information spread across gaps within paired-end reads can be utilized for generating particle trajectories. In particular, let us introduce a new variable

$\mathbf{Pos}^t = \{pos_i^t, i = 1, \cdots, m, pos_i^t \in \{0, 1, 2, \cdots, t\}\}$, where pos_i^t is the nearest informative (non-gap) position in the i^{th} row left of the column t; note that $pos_i^{t+1} = pos_i^t$ if $X_{i,t+1} = -$. Also, note that $pos_i^{t-1} = 0$ implies that there are no informative positions in the i^{th} row left of the column t (i.e., $X_{ij} = -$ for all $j \leq t-1$). With this notation, we rephrase (2.3) as

$$\begin{aligned} w_t^{(k,l)} &\propto w_{t-1}^{(k)} p_\theta(X_{\cdot t} | S_{\cdot t} = s_l, S^{(k)}_{\cdot \mathbf{Pos}^{t-1}}, X_{\cdot \mathbf{Pos}^{t-1}}) \\ &\times p_\theta(S_{\cdot t} = s_l | S^{(k)}_{\cdot \mathbf{Pos}^{t-1}}, X_{\cdot \mathbf{Pos}^{t-1}}). \end{aligned} \tag{2.4}$$

The measurement model in (2.4) assumes that the i^{th} read is randomly generated from one of the two haplotypes, i.e.,

$$p(X_{\cdot t} | S_{\cdot t} = s_l, S^{(k)}_{\cdot \mathbf{Pos}^{t-1}}, X_{\cdot \mathbf{Pos}^{t-1}}) = \prod_{i=1, x_{it} \neq -}^{m} p(X_{i,t} | S_{\cdot t} = s_l, S^{(k)}_{\cdot pos_i^{t-1}}, X_{i, pos_i^{t-1}}),$$

where

$$\begin{aligned} & p_\theta(X_{i,t} | S_{\cdot t} = s_l, S^{(k)}_{\cdot pos_i^{t-1}}, X_{i, pos_i^{t-1}}) \\ = & \begin{cases} p(x_{it} | s_{l_1}), & \text{if } X_{i, pos_i^{t-1}} = S^{(k)}_{1, pos_i^{t-1}}, \\ p(x_{it} | s_{l_2}), & \text{if } X_{i, pos_i^{t-1}} = S^{(k)}_{2, pos_i^{t-1}} \\ \frac{p(x_{it} | s_{l_1}) + p(x_{it} | s_{l_2})}{2}, & \text{otherwise.} \end{cases} \end{aligned} \tag{2.5}$$

When computing $p(S_{\cdot t} = s_l | S^{(k)}_{\cdot \mathbf{Pos}^{t-1}}, X_{\cdot \mathbf{Pos}^{t-1}})$, we assume that there is no correlation between the consecutive SNPs. Therefore, the state transition distribution is formed using composition probabilities, e.g., $p(S_{\cdot t} = s_l | S^{(k)}_{\cdot \mathbf{Pos}^{t-1}}, X_{\cdot \mathbf{Pos}^{t-1}})$ $= \Pr(S_{1t} = s_{l_1}) \Pr(S_{2t} = s_{l_2})$. Note that, in principle, side information such as genotype frequencies or the patterns of linkage disequilibrium (LD) can be incorporated in state transition probabilities.

Going back to the example illustrated in Figure 2.2, the modification of the weights shown in (2.4) now allows ParticleHap to retrieve phase information at position $t+2$ from the reads (highlighted in orange) that have a gap in position $t+1$ but cover some SNPs in the positions left of the $(t+1)^{st}$ one. This often has a beneficial effect on the switch error rate, defined as the ratio of the number of SNP positions where the two chromosomes of a resulting haplotype phase must be switched in order to reconstruct the true phase. As an illustration, consider column t in the observation matrix shown in Figure 2.2. Since none of the reads that cover SNPs at column t provide any phasing information, the partially reconstructed haplotype pair would equally likely be extended by either (C, G) or (G, C) which might lead to the switch error at the position t. This ambiguity is resolved at position $t+2$ from the reads $i_1 = 4$ and $i_2 = 13$ (highlighted in orange), which do not cover sites $t+1$, t nor $t-1$, but do cover sites 4 and 1, respectively: ParticleHap relying on those reads in (2.4)-(2.5) will assign larger weight to the particle propagated along the correct state path.

To initialize the algorithm at $t = 1$ from k possible SNPs, all possible assignments are considered as $S_{\cdot 1}^{(k)}, k = 1, \ldots, K$, with the corresponding weights $w_1^{(l)}, l = 1, \ldots, L$, computed as $w_1^{(l)} \propto p(X_{\cdot 1}|S_{\cdot 1}^{(l)} = s_l)$. From $t = 2$, all possible extensions of the $(t-1)$-long haplotype are considered, and the extensions having non-zero weights are used to generate particles until K such particles are created. Once the set of K particles is formed, the subsequently generated particles are included in the set if their weight is greater than the

weight of at least one particle that is already in the set; the latter then needs to be excluded from the set so that its cardinality remains K.

The ParticleHap algorithm is formalized below.

Step 1 (Initialization): For the first SNP position, compute $w_1^{(l)} \propto p(X_{\cdot 1}|S_{\cdot 1}^{(l)} = s_l)$, $l = 1, \ldots, L$. Normalize $w_1^{(l)}$ and store the corresponding possible haplotype pairs in $S_{\cdot 1}^{(k)}$, $k = 1, 2, \cdots, K$.

Step 2 (Run iterations for $2 \leq t \leq n$): For each step t, $t = 2, \ldots, n$, enumerate all possible extensions of the existing particles $S_{\cdot t-1}^{(k)}$, thus generating $S_{\cdot t}^{(k,l)} = [S_{\cdot t-1}^{(k)}, s_l]$, $l = 1, \cdots, L$. For all l, compute the weights $w_t^{(k,l)}$ using (2.4).

Step 3 (Particle selection): Select and store K particles $\{S_{\cdot t}^{(k)}, k = 1, \cdots, K\}$ with the highest importance weights $\{w_t^{(k)}, k = 1, \ldots, K\}$ from the set $\{S_{\cdot t}^{(k,l)}$, $w_t^{(k,l)}, k = 1, \ldots, K, l = 1, \ldots, L\}$. Normalize the weights of the selected particles. Go back to **Step 2** and repeat until $t = n$.

Step 4 (Haplotype reconstruction): At $t = n$, assemble the entire haplotype sequence by selecting the particle with the largest weight.

2.2.3 Complexity analysis of ParticleHap

Since ParticleHap searches for the most likely haplotypes by sequentially extending the partially reconstructed candidate haplotypes one position at a time, it is computationally very efficient and has complexity that scales linearly with the haplotype length, $O(n)$. On average, the amount of calculations needed for haplotype assembly with ParticleHap is nKL_aC_a, where C_a

denotes the average number of bases covering heterozygous sites and L_a denotes the average number of possible extensions of the partially reconstructed haplotype at a heterozygous SNP site. It is worth pointing out that while there are in principle 12 possible SNP pairs of (s_{l_1}, s_{l_2}) at one site, the 3^{rd} or 4^{th} most frequently reported nucleotides are never selected as potential SNPs' genotypes based on the likelihood calculations. Therefore, ParticleHap can be implemented even more efficiently by retaining the two (or three in the case of ties) most frequently observed nucleotides at each site while the others, which are considered errors, are replaced by $-$; this step can significantly reduce the amount of likelihood calculations without compromising the accuracy of computing the most likely genotype.

2.2.4 Performance analysis of ParticleHap

We evaluate the performance of ParticleHap by deriving an expression for the probability of error. Let c_j be the number of observed entries in the j^{th} column of $\mathbf{X}$, $X_{\cdot j}$. ParticleHap assigns SNPs to haplotypes sequentially, one at a time, depending on the history of the state as well as observations. Thus, the haplotyping error occurring at any given SNP position can propagate to the following sites, which leads to position dependent haplotyping error probabilities. Let $E_{\hat{S}_{\cdot j}}$ be the event that ParticleHap falsely estimates the state (haplotype phase) at the j^{th} SNP position. Then, the probability that ParticleHap recovers haplotype sequences of length n with errors at N_e SNP positions follows a Poisson binomial distribution with parameters n and

$\mathbf{P}(E_{\hat{S}_{\cdot j}})$.

$$P_e(N_e = d) = \sum_{\mathcal{D}\in\mathcal{E}_d} \prod_{i\in\mathcal{D}} \mathbf{P}(E_{\hat{S}_{\cdot i}}) \prod_{j\in\mathcal{D}^c} (1 - \mathbf{P}(E_{\hat{S}_{\cdot j}})), \tag{2.6}$$

where $\mathcal{E}_d$ is the set of all subsets of d integers selected from $\{1, \cdots, n\}$. Therefore, the probability that ParticleHap assembles haplotype sequences with errors at less than N_e SNP positions is $\sum_{d=0}^{N_e} P_e(d)$.

Let E_{c_j} denote the event where an estimate of the j^{th} state from (2.5) is wrong. Then, $E_{\hat{S}_{\cdot j}}$ happens when E_{c_j} occurs in more than half of the reads covering the j^{th} SNP position. Thus, assuming the independence of the events E_{c_j}, $\mathbf{P}(E_{\hat{S}_{\cdot j}})$ is approximated as

$$\mathbf{P}(E_{\hat{S}_{\cdot j}}) \approx \sum_{j=\lfloor \frac{c_j}{2} \rfloor + 1}^{c_j} \binom{c_j}{j} \mathbf{P}(E_{c_j})^j (1 - \mathbf{P}(E_{c_j}))^{c_j - j} + \frac{1}{2} \binom{c_j}{\frac{c_j}{2}} e^{\frac{c_j}{2}} (1 - e)^{\frac{c_j}{2}} mod(c_j, 2)$$

where e denotes sequencing error rate. As implied by (2.5), the event E_{c_j} depends on the information from both the observation and partially inferred haplotype. Thus, E_{c_j} occurs when the information from one of them is false. Let E_χ denote the event where the observation is erroneous. Then, E_χ occurs when either an observed entry in $X_{\cdot j}$ or the corresponding informative entry in the same row left of the j^{th} column (e.g., $X_{i,pos_i^{j-1}}$ in (2.5)) is erroneous due to a sequencing error. Note that if both are erroneously flipped, E_χ^c happens. Therefore,

$$\begin{aligned} \mathbf{P}(E_{c_j}) &= \mathbf{P}(E_\chi | E^c_{\hat{S}_{\cdot j-1}}) \mathbf{P}(E^c_{\hat{S}_{\cdot j-1}}) + \mathbf{P}(E^c_\chi | E_{\hat{S}_{\cdot j-1}}) \mathbf{P}(E_{\hat{S}_{\cdot j-1}}) \\ &\overset{(a)}{\approx} 2e(1 - e)(1 - \mathbf{P}(E_{\hat{S}_{\cdot j-1}})) + ((1 - e)^2 + e^2) \mathbf{P}(E_{\hat{S}_{\cdot j-1}}). \end{aligned}$$

where (a) comes from the fact that $\hat{S}_{\cdot j-1}$ depends on $X_{i,pos_i^{j-1}}$. Note that, following the derivations, the probability of error of ParticleHap in (2.6) mostly

depends on the coverage at each site $c_j, j = 1, \cdots, n$, and sequencing error rate e.

2.2.5 Postprocessing

We can further improve the accuracy of assembled haplotypes by pooling the information obtained from multiple runs of ParticleHap whose performance may be affected by the choice of the starting position. In particular, we also run the algorithm in the opposite direction, e.g., perform sequential Monte Carlo reconstruction of the most likely haplotype starting from the site $t = n$ and terminating the algorithm at the site $t = 1$. For the sites where the haplotype pairs reconstructed by the two runs of ParticleHap differ from each other, we compare the likelihoods of the solutions (i.e., we compare the weights in (2.4)) and choose the one with larger likelihood. In case the sites are consecutive, we compare MEC scores for the two cases and choose the one with smaller MEC.

2.3 Results and discussion

We implemented ParticleHap in C and compared its performance with the publicly available implementations of HapCUT [9] and ReFHap [26]. All three methods are run on a Linux OS desktop with 3.06GHz CPU and 8Gb RAM (Intel Core i7 880 processor). Both real and simulated data are used for the experiments, as described in the remainder of this section.

2.3.1 1000 Genome Project data

We first study the performance of ParticleHap on 454 sequencing data of CEU NA12878 genome (1000 Genomes Pilot Project [20]). The short-read data aligned with respect to a reference genome as well as variant and genotype calls for the individual are provided. The data contains a total of 2.58 million reads covering 1.65 million variants on 22 chromosomes. Due to the short lengths of reads and limited insert sizes, the data is split into a number of disconnected blocks. We use ParticleHap to reconstruct each such block.

To evaluate the performance of haplotype assembly, we adopt three measures: the number of phased SNPs (nPhased), the minimum error correction (MEC) score, and running time (Time). In particular, nPhased is the number of SNPs phased by a haplotype assembler; MEC score is the smallest number of entries in the data matrix which need to be changed so that the sequencing information is consistent with an error-free haplotype pair (we report the total MEC score evaluated as the sum of the MEC scores obtained for each haplotype block); and, Time is the runtime of an algorithm in seconds, measured for each algorithm on the same processor.

We choose a different number of particles K for each block. Specifically, the longer the block length n, the larger the number of particles (12 for $n \leq 12$, $\frac{n}{2}$ for $12 < n < 100$, and 50 for $n \geq 100$). We assume that the bases in the sequence are equally likely, i.e., the composition probabilities of the 4 nucleotides are equal, 0.25. Note that we assume the error rate $e = 0.01$, which is consistent with the typical sequencing accuracy of the 454 platform.

Table 2.1: The performance comparison on a CEU NA12878 data set sequenced using the 454 platform in the 1000 Genomes Project.

	ParticleHap			HapCUT			ReFHap		
chr	nPhased	MEC	Time(s)	nPhased	MEC	Time(s)	nPhased	MEC	Time(s)
1	66661	2045	1.07	66616	2293	28.03	66490	2111	5.79
2	78002	2742	1.11	77970	2857	35.71	77853	2698	7.78
3	66217	2111	3.96	66178	2349	29.50	66071	2203	6.20
4	69939	2386	3.94	69901	2591	37.16	69786	2410	9.01
5	63723	1971	4.75	63693	2156	28.06	63605	2044	6.00
6	69750	3312	5.25	69706	3544	58.60	69580	3318	13.39
7	54330	1867	3.31	54302	2059	27.23	54202	1908	7.19
8	56406	1700	3.89	56382	1828	25.87	56281	1690	5.60
9	42244	1335	2.15	42230	1472	20.02	42157	1365	4.52
10	50022	1618	2.73	49998	1814	23.44	49900	1662	5.22
11	46141	1411	2.72	46124	1569	21.66	46051	1467	4.89
12	43333	1467	2.32	43315	1581	20.00	43251	1495	3.92
13	36952	1286	0.68	36937	1398	18.80	36872	1311	7.10
14	30349	887	0.38	30334	982	13.25	30293	916	2.90
15	26626	975	0.88	26614	1055	11.46	26567	968	2.60
16	31675	1156	0.87	31662	1257	14.84	31612	1185	3.98
17	21054	1206	0.59	21048	1223	11.19	21010	1172	5.35
18	28784	851	0.37	28769	936	11.94	28717	855	2.67
19	17018	653	0.25	17006	761	8.35	16961	687	4.25
20	21679	737	0.43	21673	790	9.50	21635	735	2.84
21	14737	485	0.41	14736	525	6.82	14714	500	2.09
22	12929	388	0.28	12925	433	5.38	12891	395	1.74

A comparison of the number of phased SNPs(nPhased), the MEC scores(MEC) and running time(Time) for different haplotype assembly algorithms, ParticleHap, HapCUT and ReFHap, on all of 22 chromosomes.

Then, the sequencing error probabilities $\Pr(X_{ij} = x_{ij}|S_{kj} = s_{l_k}) = e/3$ if $x_{ij} \neq s_{l_k}$ and are equal to $1 - e$ otherwise.

Table 2.1 shows that ParticleHap can assemble more haplotypes with higher accuracy and better computational efficiency than HapCUT and ReFHap (the lower the MEC score, the better the performance). Clearly, more SNP sites are phased by ParticleHap than by either HapCUT or ReFHap (please

see columns 2, 5 and 8). This can be attributed to the fact that ParticleHap processes reads containing actual base calls (i.e., the reads represented by A, C, G, T and $-$, rather than the same reads being represented by their binary post-genotype calling counterparts), thus allowing more than two nucleotides at a site, while HapCUT and ReFHap discard some information in the process due to relying on the simplified representation of the reads (via the ternary alphabet with elements 0, 1 and $-$). It turns out that roughly $1.2 - 1.5\%$ of the heterozygous positions in the dataset are either tri- or tetra-allelic SNP sites.

As can be seen from columns 3, 6 and 9, ParticleHap outperforms HapCUT and ReFHap by consistently achieving lower MEC scores on most of the chromosomes. Note that the MEC scores are calculated only for the haplotype pairs phased by an algorithm; thus, the lower MEC of ReFHap does not necessarily imply that it achieves better performance since it may actually be phasing a smaller number of SNP sites. ParticleHap simultaneously provides longer lengths of phased haplotypes as well as lower MEC scores, demonstrating the high accuracy of the proposed algorithm (note that the total number of reads and allele calls involved in the MEC calculation of ParticleHap is larger than those for HapCUT and ReFHap). This can be partly attributed to the fact that ParticleHap works with genotype likelihoods and allows thorough examination of tri- or tetra-allele SNP sites, which leads to improved genotype accuracy in situations where there are potential errors in "hard" genotype calls used by the competing methods.

It is also worth pointing out that ParticleHap is designed to sequentially find the maximum-likelihood solution to the haplotype assembly problem rather than to optimize the MEC criterion while HapCUT uses MEC as its optimization objective (the MEC score is only used to correct potential errors in ParticleHap's post-processing step); therefore, the superior MEC performance of ParticleHap demonstrates the robustness of the approach. Finally, as reported in columns 4, 7 and 10, ParticleHap assembles haplotypes significantly faster than either HapCUT or ReFHap. In particular, ParticleHap can complete haplotype assembly for each of the 22 chromosomes within 6 seconds, while HapCUT and ReFHap require 59 and 14 seconds for the same task, respectively.

2.3.2 Simulated data

We further test the performance of our proposed method on the simulated data set. In particular, we examine how the genotype calling errors affect the performance of haplotype assembly. The data are generated using a similar strategy to the one in [30] except that a genotype calling error is included in our simulation data. We generate a pair of phased heterozygous SNP sequences of length n, which have genotype calling errors with probability g_e. The parameter g_e is judiciously chosen as 0.04 and 0.08 in order to emulate practical scenarios reported in [54] (there, the genotype call accuracy for high call rates was up to 96% with the use of LD information, from 78% and 87% with the use of single sample and multiple individuals, respectively,

for the 62 CEU individuals). The true haplotype sequences are generated by emulating genotyping errors; each base in the erroneous heterozygous SNP sequences is flipped to one of the other three nucleotides with equal probability, $g_e/3$. To generate the observation data matrix, instead of sampling (with reads) from true haplotype sequences c times as in [30], we sample true haplotype sequences $\frac{c}{2}$ times and sequences containing genotype calling errors $\frac{c}{2}$ times. Each replicate is randomly partitioned into non-overlapping fragments of length between 3 and 7 (the lengths typical of benchmarking data sets in [30]). In order to simulate paired-end (or mate-pair) sequences, we randomly merge some of the generated fragments (fragments whose SNPs are in the first half of haplotype sequence are merged with those whose SNPs in the last half of sequence; as a result, half of the fragments in the dataset are paired-end sequences). Once the fragments are arranged in a SNP matrix, we emulate sequencing errors by randomly flipping a base to one of the other three nucleotides with equal probability. The probability that each base is flipped is 0.03 and 0.01 for $g_e = 0.04$ and $g_e = 0.08$, respectively, and thus the total error rate for the entries in the SNP matrix is $e = 0.05$. To explore the performance of the algorithm over a broad range of experimental parameters, we generate datasets with different SNP lengths ($n = 100$, 200 and 300) and vary the coverage rate ($c = 4$, 6, 8 and 10) for each genotype calling error rate ($g_e = 0.04$ and $g_e = 0.08$). For each of the 24 combinations of the parameters, the experiment is repeated 100 times and the results averaged over the 100 instances are reported for each case.

Table 2.2: The performance comparison on a simulated data set for $ge = 0.04$.

		ParticleHap			HapCUT		ReFHap	
n	c	ImpGeAc	ReconRate	Time(s)	ReconRate	Time(s)	ReconRate	Time(s)
100	4	0.6254	0.9785	0.02	0.9598	0.66	0.9497	0.11
	6	0.6252	0.9794	0.02	0.9570	0.84	0.9481	0.25
	8	0.5977	0.9792	0.03	0.9590	1.01	0.9524	0.56
	10	0.5737	0.9780	0.03	0.9582	1.17	0.9517	1.23
200	4	0.5935	0.9779	0.07	0.9594	1.78	0.9499	0.26
	6	0.5977	0.9783	0.08	0.9597	2.26	0.9518	0.88
	8	0.5840	0.9757	0.09	0.9596	2.71	0.9524	2.56
	10	0.5758	0.9777	0.11	0.9593	3.13	0.9528	5.80
300	4	0.6013	0.9715	0.17	0.9591	3.39	0.9493	0.53
	6	0.5848	0.9720	0.20	0.9596	4.34	0.9511	2.12
	8	0.5842	0.9695	0.22	0.9598	5.14	0.9525	6.35
	10	0.5671	0.9703	0.24	0.9599	5.90	0.9537	14.68

A comparison of reconstruction rate(ReconRate) and running time(Time) for different haplotype assembly algorithms, ParticleHap, HapCUT and ReFHap, on the simulated data for $ge = 0.04$. For ParticleHap, the improvement rates of genotyping accuracy (ImpGeAc) are also reported.

We quantify the ability of an algorithm to reconstruct a haplotype by means of the reconstruction rate [30] defined as

$$R = 1 - \frac{\min(D(h_1, \hat{h}_1) + D(h_2, \hat{h}_2), D(h_1, \hat{h}_2) + D(h_2, \hat{h}_1))}{2l},$$

where (h_1, h_2) is the pair of true haplotypes, $(\hat{h}_1, \hat{h}_2)$ is the pair of reconstructed haplotypes, $D(h_i, \hat{h}_j) = \sum_{k=1}^{n} d(h_i[k], \hat{h}_j[k])$ is the generalized Hamming distance between h_i and $\hat{h}_j$, and $d(h_i[k], \hat{h}_j[k]) = 0$ if $h_i[k] = \hat{h}_j[k]$ and is 1 otherwise. Running time (Time(s)) for each algorithm is evaluated along with the reconstruction rate (ReconRate). In addition, we report the rate at which ParticleHap infers true genotypes for the locations where genotype calling errors are induced, i.e., the improvement rate of genotyping accuracy (labeled as ImpGeAc).

Table 2.3: The performance comparison on a simulated data set for $ge = 0.08$.

		ParticleHap			HapCUT		ReFHap	
n	c	ImpGeAc	ReconRate	Time(s)	ReconRate	Time(s)	ReconRate	Time(s)
100	4	0.6211	0.9618	0.02	0.9193	0.66	0.9009	0.11
	6	0.5941	0.9610	0.02	0.9184	0.86	0.8997	0.25
	8	0.5970	0.9585	0.03	0.9184	1.04	0.9017	0.56
	10	0.5845	0.9572	0.03	0.9193	1.19	0.9041	1.26
200	4	0.6262	0.9615	0.08	0.9186	1.80	0.8979	0.27
	6	0.5938	0.9418	0.09	0.9198	2.30	0.9021	0.89
	8	0.6050	0.9389	0.10	0.9193	2.75	0.9019	2.53
	10	0.5997	0.9438	0.12	0.9193	3.16	0.9039	5.84
300	4	0.6245	0.9432	0.18	0.9197	3.43	0.8995	0.53
	6	0.6069	0.9397	0.21	0.9198	4.49	0.9009	2.11
	8	0.6058	0.9315	0.23	0.9186	5.54	0.9009	6.29
	10	0.5792	0.9192	0.27	0.9187	6.37	0.9029	15.00

A comparison of reconstruction rate(ReconRate) and running time(Time) for different haplotype assembly algorithms, ParticleHap, HapCUT and ReFHap, on the simulated data for $ge = 0.08$. For ParticleHap, the improvement rates of genotyping accuracy (ImpGeAc) are also reported.

Tables 2.2 and 2.3 compare the results of ParticleHap, HapCUT and ReFHap for $g_e = 0.04$ and $g_e = 0.08$, respectively. As can be seen in those tables, ParticleHap assembles haplotypes with the reconstruction rates of 97.85% and 95.68% when the data is affected by the genotype calling error rates of 4% and 8%, respectively. This highly accurate performance is achieved in part due to ParticleHap's ability to improve genotyping accuracy in mis-called (or uncertain) sites by more than 50% in all the considered scenarios as shown in column 3 in Tables 2.2 and 2.3. (i.e, ParticleHap can improve the genotype accuracy of 96% and 92% to 98% and 96% in Table 2.2 and 2.3, respectively.) It is worth pointing out that, in these simulations, we assumed equal prior probabilities of all genotypes. Imposing more judicious choices of priors may

lead to further improvement of genotyping accuracy. On another note, the corresponding reconstruction rates of HapCUT and ReFHap do not exceed 96% and 92%, respectively. Evidently, incorporation of genotyping in the haplotype assembly procedure allows pushing the limits of the achievable accuracy of haplotype assembly. Note that ParticleHap is consistently much faster than HapCUT and ReFHap in all the considered scenarios. As expected, the running time of ParticleHap increases with both the haplotype length and sequencing coverage.

2.4 Summary

We presented a novel deterministic sequential Monte Carlo (i.e., particle filtering) algorithm for solving the haplotype assembly problem. ParticleHap sequentially infers the haplotype sequence, one SNP site at a time, by exhaustively searching for the most likely extension of the partially assembled haplotype in each step, examining both the possible genotypes and phase. We tested the performance of ParticleHap on 1000 Genomes Project data, showing that it achieves better minimum error correction scores and phases more heterozygous sites than two of the most accurate existing methods while being significantly more computationally efficient. The results of testing ParticleHap on the simulated dataset also demonstrate that the proposed method can reconstruct haplotypes with higher accuracy and efficiency than those of competing techniques over a wide range of the haplotype assembly problem parameters.

Chapter 3

aBayesQR: A Bayesian method for reconstruction of viral populations characterized by low diversity

3.1 Background and Related Works

A number of potentially life-threatening infectious diseases are caused by RNA viruses, including human immunodeficiency virus (HIV), hepatitis C virus (HCV), influenza and Ebola. RNA viruses have a relatively high mutation rate due to both their error-prone replication process and the lack of sophisticated repair mechanisms [24]. Consequently, they rapidly evolve and exist as a set of non-identical but closely related genetic variants, known as a viral quasispecies. Viral populations can readily adapt to dynamic environments and develop resistance to antiviral drugs and vaccines, which makes the design of effective and long-lasting treatments for RNA viral diseases exceedingly difficult [41]. Determining the structure of viral populations helps the understanding of viral diseases and provides guidance in the development of effective medical therapeutics.

Quasispecies spectrum reconstruction (QSR) aims to assemble individual haplotype sequences in a population and estimate their prevalence using sequencing reads generated from a sample containing a set of viral variants.

High-throughput next-generation sequencing (NGS) technologies have enabled affordable acquisition of data needed to assemble quasispecies. However, relatively short length of the NGS reads and the presence of errors in sequencing data render the QSR problem difficult. The QSR problem is particularly challenging when the strains in a viral population are highly similar, i.e., the sequences are characterized by low mutual genetic distances, and further exacerbated if some of those strains are relatively rare [56].

Several software tools for solving the QSR problem by analyzing NGS data have been developed in recent years. ShoRAH [78], the earliest publicly available such software, was developed by combining a path cover based approach and probabilistic clustering in [28] and [79], respectively, and applied to analysis of HIV data [80]. Read-graph approach was the basis for ViSpA [8], developed as a variant of the network flow method proposed in [76]. [58], proposed a combinatorial method for QSR and the resulting software, QuRe, was provided by [59]. An approach that resulted in the software package PredictHaplo [57] relied on a Dirichlet Process mixture model and was developed specifically targeting HIV population reconstruction; QuasiRecomb [72] is based on a hidden Markov model that explicitly models recombination events. In [66], a benchmarking study that compares the performance of several publicly available quasispecies reconstruction softwares was presented. The study demonstrated that none of the tested methods could reconstruct populations characterized by low pairwise distance between the haplotype sequences. Following this study other softwares, including HaploClique [71],

based on max-clique enumeration of a read alignment graph, and VGA [50], a graph-coloring based heuristic method, were developed. Most recently, a reference-assisted *de novo* assembly pipeline, ViQuaS, was proposed in [35]. ViQuaS extends an existing algorithm, QuRe [58], and outperforms various other techniques on a wide range of dataset. However, performance of these more recent methods deteriorates dramatically in the scenarios where the genetic diversity of a population is low [56].

Both [56] and [66] have pointed out that the existing methods for viral quasispecies reconstruction struggle in the scenarios where the populations are characterized by low diversity. This, in part, is due to the presence of relatively long genetic regions that are common to pairs of closely related viral sequences; clearly, this makes distinguishing different strains challenging. The problem becomes even more difficult when the frequency of one (or more) of the close strains is low; in such settings small genetic distances may be confused for sequencing errors and hence remain undetected. Such failures to detect may have serious consequences in antiviral treatment studies since undetected strains cannot be properly targeted for drug and vaccine design. It has been shown that even the viral strains existing at low frequencies can cause a drug treatment failure due to their resistance to the drug [42, 69]. Therefore, complete recovery of the composition of viral populations is of critical importance for effective antiviral therapies.

In this chapter, we present a novel QSR algorithm, aBayesQR (combining agglomerative hierarchical clustering and Bayesian inference), that over-

comes limitations of the existing methods and reliably reconstructs quasispecies characterized by low diversity [5][1]. The algorithm performs reconstruction of a quasispecies from next-generation sequencing (NGS) data in two stages. In the first stage, conflict-free short reads are hierarchically merged and assembled into longer sequences (contigs) which we refer to as super-reads. In the second stage, likelihoods of the probable quasispecies are computed using the assembled super-reads (rather than using the original set of short reads), and the most likely set of viral strains is selected. Note that the super-reads synthesized in the first stage of aBayesQR allow us to distinguish between closely related strains which share long genetic regions as well as reduce the search space and enable computational tractability of the Bayesian inference conducted in the second stage. The second stage of aBayesQR involves sequential pruning of the solution space; in particular, the likely set of partial viral strains comprising n single nucleotide variants (SNVs) is generated by extending previously inferred partial viral strains having $n-1$ SNVs. The number of sequences in a set (i.e., the size of a viral population) is dynamically updated at each step by evaluating quality of the set of partially reconstructed viral strains, and ultimately precisely inferred at the end of the search process. The relative frequencies of each strain are determined by counting the numbers of reads unambiguously associated with each of the reconstructed strains. Our

[1]This work has been published as [Soyeon Ahn and Haris Vikalo. abayesqr: A bayesian method for reconstruction of viral populations characterized by low diversity. In International Conference on Research in Computational Molecular Biology, pages 353-369. Springer, 2017]. The author of this dissertation is the primary contributor.

tests on both simulated and experimental data demonstrate superior performance compared to state-of-the-art methods for quasispecies reconstruction. In particular, it is shown that unlike the competing methods, aBayesQR is capable of detecting and reliably reconstructing viral haplotypes having very small mutual genetic distances.

3.2 Method

Our algorithm for inferring spectrum of a viral population consists of the following two steps: (1) constructing super-reads by hierarchically clustering aligned paired-end reads, (2) inferring the most likely quasispecies from the set of super-reads and estimating the frequencies of the strains in the quasispecies.

3.2.1 Super-reads construction via agglomerative clustering

In the first stage of aBayesQR, paired-end reads uniquely mapped to a reference genome are grouped into super-reads via agglomerative hierarchical clustering. This is facilitated by a weighted graph $G = (\mathcal{V}, \mathcal{E})$ which is constructed and recursively updated as the clustering proceeds. In particular, each vertex of G is associated with a cluster collecting reads that originated from a single strain in a quasispecies; we denote the set of reads in the i^{th} cluster (i.e., the cluster associated with the i^{th} vertex) as $V_i = \{v_i^j, j = 1, \cdots, |V_i|\}$. Let sr_i denote a consensus sequence (i.e., a super-read) constructed from the reads in V_i. The i^{th} and j^{th} vertex of G are connected by an edge $e_{ij} \in \mathcal{E}$ if

all the reads in V_i and V_j (or, equivalently, sr_i and sr_j) are conflict-free and an overlap criterion, specified later in this section, is satisfied. The weight w_{ij} of the edge e_{ij} is a measure of similarity between V_i and V_j at each step, the algorithm merges a pair of vertices connected by the edge having the largest weight to form a new vertex and agglomerates the corresponding clusters.

The alleles at homozygous sites, common to all the components of a quasispecies, are not utilized in the reconstruction procedure. Instead, we separate reads having originated from different strains by clustering them using heterogeneous sites with reliable SNV information. An SNV information is considered reliable if the relative abundance of the allele is above a pre-determined threshold, as in [23]; alleles whose abundance is below the threshold are treated as sequencing errors and disregarded in the process of clustering. For convenience, let us denote the set of pre-processed paired-end reads by $R = \{r_i, i = 1, \cdots, |R|\}$. The agglomerative clustering is initialized with $|R|$ clusters, one for each read; in other words, we start with $V_1 = r_1, \cdots, V_{|R|} = r_{|R|}$, implying that $|\mathcal{V}| = \sum_{i=1}^{|\mathcal{V}|} |V_i| = |R|$, and proceed by sequentially merging judiciously chosen pairs of vertices (i.e., agglomerating the corresponding clusters). Intuitively, it is meaningful to reduce the number of vertices in the graph by merging those associated with conflict-free consensus sequences that have a large overlap. To formalize this, let $L_i = \{l_1, \cdots, l_{|L_i|}\}$ denote an index set of the SNV positions covered by sr_i, let $L_{i\cap j} = \{l_1, \cdots, l_{|L_{i\cap j}|}\}$ be the index set of SNV positions covered by both sr_i and sr_j, and let $L_{i\cup j} = \{l_1, \cdots, l_{|L_{i\cup j}|}\}$ be the index set of SNV positions

covered by either sr_i or sr_j. Then the pairs of vertices (i, j) that we consider as candidates for merging and thus connect by an edge are those satisfying either

$$|L_{i\cap j}| \geq \theta \cdot |L_{i\cup j}| \quad \text{or} \quad |L_{i\cap j}| = \min(|L_i|, |L_j|),$$

where the 2^{nd} condition promotes merger of short super-reads, and the choice of θ is discussed below. To quantify uncertainty inherent to a clustering solution due to existence of non-overlapping positions among the reads in each cluster, we define a position-specific confidence score

$$score_i[l] = \frac{cr_i[l] - cr[l]}{1 - cr[l]}$$

where l denotes the position, $cr[\cdot]$ is the overall coverage rate, and $cr_i[\cdot]$ denotes cluster-specific coverage rate for V_i (i.e., $cr_i[l]$ is the fraction of reads in $V_i = \{v_i^j, j = 1, \cdots, |V_i|\}$ covering position l). On the one hand, this score is penalized at a site where the fraction of cluster members (short reads) covering the site is low; the score is negative if the cluster-specific coverage rate is below the global coverage rate which implies uncertainty of the clustering decision. On the other hand, positive scores indicate high confidence in the decision to group the reads into the same cluster. Note that the highest possible score of 1 at position l is achieved when all the reads in a cluster cover the l^{th} position. Using the confidence scores, we define the weight w_{ij} assigned to an edge e_{ij} to quantify similarity between V_i and V_j as

$$w_{ij} = \frac{1}{|L_{i\cup j}|} \sum_{l \in L_{i\cup j}} score_{i\cup j}[l].$$

Algorithm 1: Agglomerative clustering for super-reads construction

Input: Set of reads aligned to the reference genome
Output: Set of super-reads and the corresponding confidence scores
for $\theta > 0$ **do**
    Build a weighted graph $G = (\mathcal{V}, \mathcal{E})$
    while E $\neq \emptyset$ **do**
        Merge two clusters connected with the largest weight
        Update $G = (\mathcal{V}, \mathcal{E})$ and weights using partial maximum array
    end while
    $\theta = \theta - 0.1$
end for

Given the weights w_{ij}, we can now specify the clustering procedure. In each step, the pair of vertices connected by the edge with maximum weight is merged; the newly constructed vertex inherits edges from the merged vertices and the weights on those edges are re-evaluated. A new (longer) consensus sequence is constructed by combining the two super-reads associated with the merged vertices; recall that there are no conflicts between the super-reads being merged. If after such an update step no edges connect the new vertex with the rest of the graph (because no inherited edges satisfy the connectivity condition), θ is decreased and the above process is repeated. We initially set θ to 0.9 and gradually decrease it by 0.1 while $\theta > 0$. The above procedure is repeated until no pairs of vertices satisfy the connectivity condition. By that point, a set of long consensus sequences (the final super-reads) has been formed from the clusters of reads associated with the nodes of the final graph. While the complexity of agglomerative clustering is, in general, $O(N^3)$ where N denotes the input data size [64], it has been shown that its time complexity can

be reduced to $O(N^2)$ with accuracy equal to that of the brute-force method by using the partial maximum array technique [36]. We exploit this to efficiently construct super-reads. The algorithm for super-read construction is formalized as Algorithm 1.

3.2.2 ML reconstruction of quasispecies from super-reads

Here we describe how to reconstruct the most likely set of strains in a viral quasispecies using super-reads from Section 3.2.1 and their confidence scores. While in principle the method outlined in this section could be applied directly to the short reads provided by a sequencing platform, such an approach would in general not only be computationally prohibitive due to a very large number of short reads but also limit the ability of the algorithm to distinguish strains with small mutual genetic distances due to having long conserved regions. Relying on a relatively small number of long super-reads constructed from short reads circumvents both of these problems and makes the reconstruction more accurate and practically feasible. Note that sequencing errors may undesirably prevent clusters of reads from being merged with other clusters due to a violation of conflict-free requirement; consequently, a set of short reads in a small cluster is likely to have a disproportionate amount of sequencing errors. For this reason, we ignore clusters with very small memberships (in particular, those containing fewer than $0.001 \cdot |R|$ reads), which limits the detection of strains to those constituting more than 0.1% of the quasispecies.

Let $\mathcal{C} = \{C_m, m = 1, \cdots, M\}$ denote the collection of clusters that remain after deleting clusters having only few reads; moreover, for convenience let us re-label the reads in C_m as c_m^j, i.e, $C_m = \{c_m^j, j = 1, \cdots, |C_m|\}$ where $c_m^j \in R$. We organize the super-reads obtained by Algorithm 1 in Section 3.2.1 into the rows of an $M \times N$ matrix $\mathbf{S} = \{s_{mn}, m = 1, \cdots, M, n = 1, \cdots, N\}$ with entries $s_{mn} \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}, -\}$ where $-$ denotes a site not covered by a super-read and N denotes the total number of SNV sites in the strains of a quasispecies. A nucleotides in the (m, n) position of $\mathbf{S}$ is assigned confidence $score_m[n]$ defined in Section 3.2.1; the scores for the entire matrix are normalized so that they fall between 0 and 1 in order to use them in our Bayesian approach to assembly. Let ε_{mn} be the probability that s_{mn} was estimated erroneously due to either a sequencing error in reads on the n^{th} SNV position or the uncertainty induced by reads not covering the n^{th} SNV position. Note that negative scores indicates low confidence resulting from insufficient cluster-specific coverage rate while positive scores imply relatively confident information. In order to map $score_m[n] \in (-\infty, 1]$ to the set $[0, 1]$, we set $\varepsilon_{mn} = 1 - e^{score_m[n]}$ for $score_m[n] < ln(1 - \epsilon)$, where ϵ denotes the error rate of a sequencing platform. Otherwise, we set $\varepsilon_{mn} = \epsilon$.

Let $Q = \{q_k, k = 1, \cdots, K\}$ denote the set of K strains of a viral quasispecies. The goal in the second stage of our method is to determine Q from the super-reads matrix $\mathbf{S}$ using a probabilistic framework. An exhaustive search over the entire solution space is computationally intractable even for small $\mathbf{S}$; instead, we reconstruct the set of K viral strains sequentially, extending

partially estimated strains one SNV position at each step. Since maintaining and extending all possible partial strains inevitably increases their number exponentially, unlikely sets of candidate strains are pruned in each step. Each step consists of three basic parts: (a) extension of the partially reconstructed strains, (b) selection of probable sets comprising K strains chosen among those generated in step (a), and (c) evaluation of the quality of the selected sets of strains and an update of K. The sequential Bayesian inference procedure in step t is illustrated in Figure 3.1.

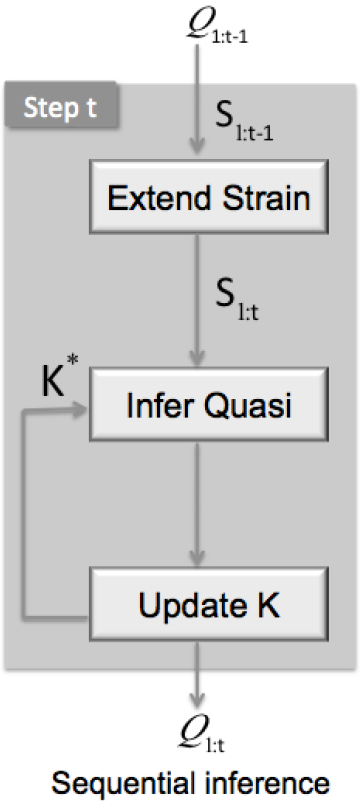

Figure 3.1: Procedure of sequential Bayesian inference in step t

Extending partially reconstructed strains. Let $F_{1:t\text{-}1} = \{f^i_{1:t\text{-}1}, i = 1, \cdots, |F_{1:t\text{-}1}|\}$ be the collection of partially reconstructed strains covering the first $t-1$ SNV sites and let $B_t = \{b^j_t, j = 1, \cdots, |B_t|\}$ be the lists of distinct bases in the t^{th} column of $\mathbf{S}$, where $b^i_t \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$ and $2 \leq |B_t| \leq 4$. Then, all the possible extensions of $f^i_{1:t-1}$ to the SNV site t can be enumerated as $\{[f^i_{1:t\text{-}1}, b^1_t], \cdots, [f^i_{1:t\text{-}1}, b^{|B_t|}_t]\}$. Let $S^i_{1:t\text{-}1} = \{s^{i_{c'}}_{1:t\text{-}1}, c' = 1, \cdots, |S^i_{1:t\text{-}1}|\}$ be the collection of super-reads covering some of the first t SNV sites which are consistent with $f^i_{1:t\text{-}1}$ (ignoring "$-$" in $s^{i_{c'}}_{1:t\text{-}1}$) where $\{i_{c'}\}$ denote indices of rows of $\mathbf{S}$ that are placed in $S^i_{1:t\text{-}1}$, and let $S^i_t = \{s^{i_c}_t, c = 1, \cdots, |S^i_t|\}$ denote the collection of nucleotides ($s^{i_c}_t \in \{\mathsf{A}, \mathsf{C}, \mathsf{G}, \mathsf{T}\}$, not "$-$") observed at the t^{th} SNV site of the super-reads in $S^i_{1:t\text{-}1}$ where $\{i_c\}$ denote the indices of rows in $\mathbf{S}$ that contribute to S^i_t. Given $S^i_{1:t\text{-}1}$, S^i_t and $f^i_{1:t\text{-}1}$, the probability of b^j_t being the true extension of $f^i_{1:t\text{-}1}$ is given by

$$P(S^i_t | b^j_t, S^i_{1:t\text{-}1}, f^i_{1:t\text{-}1}) = \prod_{c=1}^{|S^i_t|} P(s^{i_c}_t | b^j_t),$$

$$P(s^{i_c}_t | b^j_t) = \begin{cases} 1 - \varepsilon_{i_c t}, & \text{if } b^j_t = s^{i_c}_t, \\ \frac{\varepsilon_{i_c t}}{|B_t|}, & \text{otherwise.} \end{cases}$$

We extend $f^i_{1:t\text{-}1}$ to $[f^i_{1:t\text{-}1}, b^j_t] \in F_{1:t\text{-}1,t}$ by appending the $b^j_t \in B_t$ which satisfies $\frac{P(S^i_t | b^j_t, S^i_{1:t\text{-}1}, f^i_{1:t\text{-}1})^{\frac{1}{|S^i_t|}}}{\sum_{B_t} P(S^i_t | b^j_t, S^i_{1:t\text{-}1}, f^i_{1:t\text{-}1})^{\frac{1}{|S^i_t|}}} \geq \delta_0$, where the exponent ensures proper normalization and is needed since the number of super-reads, $|S^i_{1:t\text{-}1}|$, varies for each $\{f^i_{1:t\text{-}1}, i = 1, \cdots, |F_{1:t\text{-}1}|\}$. For $f^i_{1:t\text{-}1}$ which has no matched super-reads, i.e., $|S^i_{1:i\text{-}1}| = 0$, we keep all of $|B_t|$ possible extensions of $f^{i\text{-}1}_{1:t}$. By collecting probable extensions

for each $f^i_{1:t\text{-}1} \in F_{1:t\text{-}1}$, we obtain the set of partial strains stretching over the first t SNV sites, $F_{1:t\text{-}1,t}$. This procedure is formalized as function *ExtendFrag*.

---

function *ExtendFrag*$(F_{1:t\text{-}1}, t, \delta_0)$: Extend $F_{1:t\text{-}1}$ to $F_{1:t\text{-}1,t}$
Input: $F_{1:t\text{-}1}, t, \delta_0$
Output: $F_{1:t\text{-}1,t}$
for $f^i_{1:t\text{-}1} \in F_{1:t\text{-}1}$ **do**
  for $b^j_t \in B_t$ **do**
    if $\frac{P(S^i_t|b^j_t,S^i_{1:t\text{-}1},f^i_{1:t\text{-}1})^{\frac{1}{|S^i_t|}}}{\sum\limits_{B_t} P(S^i_t|b^j_t,S^i_{1:t\text{-}1},f^i_{1:t\text{-}1})^{\frac{1}{|S^i_t|}}} \geq \delta_0$ **then**
      $F_{1:t\text{-}1,t} \leftarrow F_{1:t\text{-}1,t} \cup \{[f^i_{1:t\text{-}1}, b^j_t]\}$
    end if
  end for
end for

Inferring likely sets of K partial strains. Having generated the probable partial strains $F_{1:t\text{-}1,t}$, we denote the set of all its possible subsets of K strains (i.e., the quasispecies population candidates) as $\Omega_{1:t\text{-}1,t} = \{Q^i_{1:t\text{-}1,t}, i = 1, \cdots, \binom{|F_{1:t\text{-}1,t}|}{K}\}$ where $Q^i_{1:t\text{-}1,t} = \{q^i_{kn}, k = 1, \cdots, K, n = 1, \cdots, t\}$ and $q^i_{kn} \in F_{1:t\text{-}1,t}$. The log-likelihoods of $Q^i_{1:t\text{-}1,t}$ can be expressed as

$$\ln P(\mathbf{S}|Q^i_{1:t\text{-}1,t}) = \sum_{m=1}^{M} \ln P(s_{m\cdot}|Q^i_{1:t\text{-}1,t}),$$
$$P(s_{m\cdot}|Q^i_{1:t\text{-}1,t}) = \frac{1}{K}\left(\sum_{k=1}^{K}\left(\prod_{n=1}^{t} P(s_{mn}|q^i_{kn})\right)\right),$$

where $s_{m\cdot}$ denotes the m^{th} row vector of the matrix of super-reads $\mathbf{S}$ and

$$P(s_{mn}|q^i_{kn}) = \begin{cases} 1-\varepsilon_{mn}, & \text{if } q^i_{kn} = s_{mn}, \\ \frac{\varepsilon_{mn}}{|B_n|}, & \text{if } q^i_{kn} \neq s_{mn} \end{cases} \quad \text{for} \quad s_{mn} \neq -.$$

function *InferQuasi*$(F_{1:t\text{-}1,t}, K, \delta_1)$: Infer likely sets of K strains $\mathcal{Q}_{1:t}$ from $F_{1:t\text{-}1,t}$
Input: $F_{1:t\text{-}1,t}, K, \delta_1$
Output: $\mathcal{Q}_{1:t}$
Enumerate $\mathcal{Q}_{1:t\text{-}1,t}$ from $F_{1:t\text{-}1,t}$ and K
for $Q^i_{1:t\text{-}1,t} \in \mathcal{Q}_{1:t\text{-}1,t}$ **do**
    if $P(\mathbf{S}|Q^i_{1:t\text{-}1,t}) > \delta_1 \cdot Q^{max}_{1:t}$ **then**
        $\mathcal{Q}_{1:t} \leftarrow \mathcal{Q}_{1:t} \cup \{Q^i_{1:t\text{-}1,t}\}$
    end if
end for

Let $Q^{max}_{1:t} = \max_{Q^i_{1:t\text{-}1,t} \in \mathcal{Q}_{1:t\text{-}1,t}} P(\mathbf{S}|Q^i_{1:t\text{-}1,t})$. Among the $\binom{|F_{1:t\text{-}1,t}|}{K}$ sets in $\mathcal{Q}_{1:t\text{-}1,t}$, we keep only those that satisfy $P(\mathbf{S}|Q^i_{1:t\text{-}1,t}) > \delta_1 \cdot Q^{max}_{1:t}$ while the others are discarded; let us denote the collection of candidate sets that pass this test as $\mathcal{Q}_{1:t}$. For practical feasibility of the scheme, the collection of partially reconstructed strains $F_{1:t\text{-}1,t}$ is trimmed by excluding from it all the strains that are not part of at least one of the sets in $\mathcal{Q}_{1:t}$; we denote the resulting collection of partial strains by $F_{1:t} \in F_{1:t\text{-}1,t}$ and use it when extending the strains onto the $t+1$ SNV site. The described procedure is formalized as function *InferQuasi*.

Determining the number of strains K in a quasispecies. In this step, we assess appropriateness of K used in the inference of $\mathcal{Q}_{1:t}$ and update it if necessary. To this end, we rely on the minimum error correction (MEC) score which has previously been broadly used as a criterion in the design of methods for haplotype assembly [39, 49]. In the context of polyploid haplotype assembly, the MEC score is defined as the smallest number of nucleotides that needs to be changed in data (i.e., in observed reads) so that the corrected reads are consistent with having originated from K haplotypes. Let $HD_t(\cdot,\cdot)$

denote the Hamming distance between two sequences counted over the observed nucleotides in the first t SNV positions.[2] Then the MEC score of the most likely set $Q_{1:t}^{max}$of K viral strains evaluated on the first t SNVs is

$$\mathrm{MEC}_t(K) = \sum_{m=1}^{M} \min_{k\in\{1,\cdots,K\}} \sum_{j=1}^{|C_m|} HD_t(c_m^j, q_{k\cdot}^{max}),$$

where $q_{k\cdot}^{max}$ is the k^{th} row vector of $Q_{1:t}^{max}$. Let N_t be the total number of nucleotides observed in the first t SNV positions of all the reads of the dataset. Note that the smaller the MEC scores, the higher the accuracy of a clustering. If $\mathrm{MEC}_t(K)/N_t < 2\epsilon$, we use the same value K in the next step where the likely set of viral strains stretching over the first $t+1$ SNV positions is inferred. Otherwise, we increase K by 1, repeat the estimation of $\mathcal{Q}_{1:t}$, and evaluate the improvement rate of MEC score as

$$MECimpr(K) = \frac{\mathrm{MEC}_t(K) - \mathrm{MEC}_t(K+1)}{\mathrm{MEC}_t(K)}.$$

The reason for selecting K based on the MEC improvement rate ($MECimpr$) is that the MEC score drops significantly once K matches the actual number of clusters; our scheme attempts to detect that change in order to infer population size. If $MECimpr(K) > \eta$, where η denotes a pre-specified threshold, the number of species is updated as $K \leftarrow \min\{K+n, |F_{1:t\text{-}1,t}|\}$ where n is the smallest integer number such that $MECimpr(K+n) < \eta$. If $MECimpr(K) < \eta$, we update the number of species as $K \leftarrow \max\{K-n, 2\}$ where n is the

[2]If either of the two sequences has a gap "−" in a position, that position is ignored in the computation of the aforementioned Hamming distance.

Algorithm 2: Sequential Bayesian Inference for quasispecies reconstruction

Input: Set of super-reads and the corresponding confidence scores
Output: Set of K strains of a viral quasispecies
Initial $K \leftarrow 2$, $F_{1:1} \leftarrow B_1$
for t $\in \{2, \cdots, N\}$ **do**
    $F_{1:t\text{-}1,t} = \boldsymbol{ExtendFrag}(F_{1:t\text{-}1}, t, \delta_0)$
    $\mathcal{Q}_{1:t} = \boldsymbol{InferQuasi}(F_{1:t\text{-}1,t}, K, \delta_1)$
    $K^* \leftarrow K$, $\mathcal{Q}^*_{1:t} \leftarrow \mathcal{Q}_{1:t}$
    if $\mathrm{MEC}_t(K)/N_t \geq 2\epsilon$ and $K < |F_{1:t\text{-}1,t}|$ **do**
        $\mathcal{Q}_{1:t} = \boldsymbol{InferQuasi}(F_{1:t\text{-}1,t}, K{+}1, \delta_1)$
        if $MECimpr(K) < \eta$ **do**
            while $MECimpr(K) < \eta$ and $K > 2$
                $\mathcal{Q}^*_{1:t} \leftarrow \mathcal{Q}_{1:t}$, $K^* \leftarrow K$, $K \leftarrow K - 1$
                $\mathcal{Q}_{1:t} = \boldsymbol{InferQuasi}(F_{1:t\text{-}1,t}, K, \delta_1)$
            end while
        else do
            while $MECimpr(K) \geq \eta$ and $K < |F_{1:t\text{-}1,t}|$
                $\mathcal{Q}^*_{1:t} \leftarrow \mathcal{Q}_{1:t}$, $K^* \leftarrow K$
                $\mathcal{Q}_{1:t} = \boldsymbol{InferQuasi}(F_{1:t\text{-}1,t}, K{+}1, \delta_1)$
            end while
        end if
    end if
    $K \leftarrow K^*$, $\mathcal{Q}_{1:t} \leftarrow \mathcal{Q}^*_{1:t}$
    Get $F_{1:t}$ by pruning $F_{1:t\text{-}1,t}$ based on $\mathcal{Q}_{1:t}$
end for
Reconstruct full-length quasispecies Q from $Q^{max}_{1:N} \in \mathcal{Q}_{1:t}$ and R
Estimate frequencies of each strain $q_k \in Q$ based on $HD(sr_i, q_k)$ and $|C_i|$

smallest integer such that $MECimpr(K - n) \geq \eta$. The choice of threshold η is discussed in the Appendix A. The updated value of K is used for the inference of $\mathcal{Q}_{1:t+1}$. Note that the probable set of viral strains, $\mathcal{Q}_{1:t}$, is stored for each K to avoid performing redundant $MECimpr(\cdot)$ calculations.

Once we obtain the most likely set of K viral sequences covering N

SNVs, $Q_{1:N}^{max}$, the full-length K quasispecies strains are reconstructed by inserting the consensus nucleotides observed in R into the non-SNV sites. We estimate relative frequencies p_k, $1 \leq k \leq K$, of quasispecies strains based on the Hamming distance between super-reads and the reconstructed sequences. In particular, for each super-read sr_i we determine the nearest assembled strain q_j where $j = \arg \min_{k \in \{1,\cdots,K\}} HD(sr_i, q_k)$ and the number of reads involved in constructing the super-read sr_i is counted towards p_j. The entire scheme proposed in this section is summarized as Algorithm 2.

3.3 Results and Discussion

3.3.1 Performance comparison on simulated data

To evaluate performance of the proposed method for quasispecies reconstruction, we use metrics *Recall*, *Precision*, *Predicted Proportion*, and *Reconstruction Rate*. *Recall* is defined as the ratio of the number of correctly reconstructed strains to the total number of true strains in the quasispecies, i.e., $Recall = \frac{TP}{TP+FN}$, while *Precision* is defined as the fraction of correctly reconstructed strains among all the assembled sequences, i.e., $Precision = \frac{TP}{TP+FP}$. Noting that *Precision* usually reports high scores when the number of strains is underestimated while penalizing overestimation of the population size, we also report the ratio of the number of reconstructed sequences to the true population size, *Predicted Proportion*. The closer *Predicted Proportion* to 1, the more accurate the number of reconstructed strains. Moreover, to assess

the degree of reconstruction accuracy, we define

$$\textit{Reconstruction Rate} = \frac{1}{K}\sum_{k=1}^{K}\left(1 - \frac{HD(q_k, \hat{q_k})}{G}\right),$$

where G is the length of a genome, K is the number of strains in a quasispecies and q_k and $\hat{q_k}$ denote the k^{th} true strain and its nearest sequence among the K estimated ones, respectively. To assess the accuracy of estimated frequencies, we use Jensen-Shannon divergence (JSD) which quantifies similarity between two distributions. Given a true distribution P and its approximation Q, the Kullback-Leibler (KL) divergence

$$D(P||Q) = \sum_{i=1}^{n} P(i)\log\frac{P(i)}{Q(i)}$$

is undefined when $Q(i) = 0$. JSD, a symmetrized and smoothed version of the KL divergence, circumvents this problem by defining similarity of P and Q as

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M),$$

where M is defined as $M = \frac{1}{2}(P + Q)$.

We compare our algorithm with publicly available ShoRAH [78], PredictHaplo [57], and ViQuaS [35]. Since ViQuaS is an extension of the algorithm in [58, 59], and was shown to have superior performance compared to its predecessor, we omit the comparison with the software QuRe in [58, 59]. It is worth pointing out that for the synthetic data sets we study, ShoRAH could not reconstruct strains in the regions where the simulated sequencing coverage is relatively low compared to the average, resulting in reconstruction of strains

that are shorter than the true length G. To facilitate a fair comparison with ShoRAH, we aligned its reconstructed strains to the reference genome and completed missing sites with bases from the reference. ViQuaS, on the other hand, tends to reconstruct many more strains than actually present; thus we followed ViQuaS's authors recommendation and retained only those having frequencies greater than f_{min} when calculating *Precision*. Finally, not all of the synthetic data sets could be processed with PredictHaplo, preventing us from reporting its performance in some of the scenarios.

We generated synthetic datasets by emulating high-throughput sequencing of a viral population consisting of a number of closely related viral genomes having length of 1300bp; this particular length was chosen to coincide with the longest region of the HIV *pol* gene. Quasispecies sequences are generated by introducing independent mutations at uniformly random locations along the length of a randomly generated reference genome so as to obtain a predefined level of diversity ($div\%$), i.e., a predefined average Hamming distance between quasispecies strains. Simulating Illumina's MiSeq data, 2×250bp-long paired-end reads are sampled uniformly from each viral strain with a mean coverage of $cov\times$ per strain. Inserts of the paired-end reads are on average 150bp long with standard deviation of 30. In our benchmarking tests, we focus on exploring the effects of diversity ($div\%$) on the accuracy of the quasispecies reconstruction. Two sets of viral populations are considered: (1) a mix of 5 viral strains with abundance levels $50\%, 30\%, 15\%, 4\%$ and 1%; and (2) a mix of 10 strains with abundance levels $36\%, 24\%, 16\%, 8\%, 5.5\%, 4\%, 3\%, 2\%, 1\%$

and 0.5%. Note that the abundances are chosen to approximately follow geometric distribution and that the populations include low abundant strains. For each combination of the parameters, 100 data sets were generated and the reported results were obtained by averaging over those data instances. For PredictHaplo, which did not produce results in each instance, the averaged results are reported if more than 50 instances were successfully processed.

In all of the following experiments, potential SNVs are called if their abundance is higher than 1%, which is set relatively high to avoid false positives (FPs); FPs prevent reads to be merged with existing clusters in Section 3.2.1. We execute the function ***ExtendFrag*** with parameter $\delta_0 = 0.1$. Parameter δ_1 in function ***InferQuasi*** is initially set to 0.001, but adaptively increases if the number of combinations of partially reconstructed strains exceeds 10000; this is done to limit the number of likelihood calculations performed in each run of ***InferQuasi***. The recommended value of η, a threshold used to determine population size K based on *MECimpr(·)*, is discussed in Appendix A.

We compare performances of aBayesQR, ShoRAH, ViQuaS and PredictHap when applied to the reconstruction of a quasispecies spectrum with diversity levels varying between 1% and 5% (i.e., $div \in \{1\%, 2\%, 3\%, 4\%, 5\%\}$). To test the ability of different methods to reconstruct quasispecies with low diversity, we assume low sequencing error rate of $err = 0.1\%$ (median mismatch error rates for 454 Life Sciences and Illumina platforms are 0.1% and 0.12%, respectively [6]). Coverage per strain $cov\times$ is set to $500\times$, implying total coverage of $2500\times$ and $5000\times$ for the 5-strain and 10-strain population,

Table 3.1: Performance comparison of different methods for varied diversities (div) on simulated data.

		5 strains					10 strains				
	div(%)	1	2	3	4	5	1	2	3	4	5
Recall	aBayesQR	**0.7080**	**0.7120**	**0.6840**	0.6560	0.6320	**0.5810**	**0.6380**	**0.6080**	**0.5860**	**0.5550**
	ShoRAH	0.1920	0.1600	0.1300	0.1060	0.0780	0.0150	0.0380	0.0740	0.0640	0.0930
	ViQuaS	0.3700	0.5240	0.6040	0.6360	0.5960	0.0980	0.1700	0.3730	0.4720	0.5050
	PredictHaplo	-	-	-	**0.6918**	**0.6808**	-	-	0.1021	0.1550	0.2010
Precision	aBayesQR	**0.7113**	**0.7130**	**0.6826**	0.6447	0.6319	**0.6210**	**0.6881**	**0.6610**	**0.6373**	0.6140
	ShoRAH	0.1062	0.1418	0.1240	0.1078	0.0790	0.0050	0.0170	0.0498	0.0506	0.0824
	ViQuaS	0.1960	0.3206	0.4559	0.4982	0.5298	0.0485	0.1079	0.2973	0.4690	0.5596
	PredictHaplo	-	-	-	**0.9373**	**0.8822**	-	-	0.4509	0.6000	**0.6833**
PredProp	aBayesQR	**1.0180**	**1.0120**	**1.0120**	**1.0360**	**1.0140**	**0.9680**	**0.9440**	**0.9240**	**0.9240**	**0.9100**
	ShoRAH	1.9660	1.2200	1.0780	1.0000	1.0180	3.2000	2.9100	1.6710	1.3520	1.1860
	ViQuaS	2.1100	1.7220	1.4080	1.3340	1.2180	2.0860	1.8580	1.5450	1.2320	1.0730
	PredictHaplo	-	-	-	0.7388	0.7737	-	-	0.1947	0.2430	0.2890
ReconRate	aBayesQR	**0.9990**	**0.9982**	**0.9971**	**0.9961**	**0.9953**	**0.9975**	**0.9967**	**0.9952**	**0.9942**	**0.9924**
	ShoRAH	0.9948	0.9903	0.9891	0.9851	0.9827	0.9941	0.9900	0.9899	0.9897	0.9911
	ViQuaS	0.9963	0.9949	0.9917	0.9936	0.9897	0.9944	0.9910	0.9899	0.9881	0.9858
	PredictHaplo	-	-	-	0.9906	0.9896	-	-	0.9850	0.9797	0.9747
JSD	aBayesQR	**0.0022**	**0.0008**	**0.0008**	0.0014	**0.0008**	**0.0043**	**0.0026**	**0.0023**	**0.0023**	**0.0025**
	ShoRAH	0.0762	0.0174	0.0047	**0.0009**	0.0012	0.1390	0.1110	0.0422	0.0238	0.0109
	ViQuaS	0.0651	0.0255	0.0222	0.0097	0.0180	0.0993	0.0747	0.0495	0.0469	0.0454
	PredictHaplo	-	-	-	0.1020	0.1036	-	-	0.1971	0.1636	0.1312

Performance comparison of aBayesQR, ShoRAH, ViQuaS and PredictHaplo in terms of *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD* on the simulated data with $err = 0.1\%$ and $cov = 500\times$ vs. div for a mixture of 5 and 10 viral strains. Averaged PredictHaplo results are reported if it provides answers for more than 50% of data sets. Boldface values indicate the best performance for each div(%).

respectively; strains having frequencies 0.23% or higher in the 5-strain case and those with frequencies 0.46% or higher in the 10-strain case are covered with probability 0.99 [28].

Table 3.1 demonstrates that the proposed aBayesQR algorithm outperforms existing schemes. In terms of *Recall* and *Precision*, aBayesQR exhibits exceptionally good performance compared to competing methods when recon-

structing quasispecies strains with diversity $div < 4\%$. The performance of ViQuaS deteriorates at low diversities in terms of most of the criteria (i.e., *Recall*, *Precision*, *Predicted Proportion* and *JSD*). PredictHaplo could not perform reconstruction in most of the low diversity instances yet it overall achieves the highest *Precision* because it typically underestimates the number of strains as shown by *Predicted Proportion* (e.g., estimating only 2-3 out of 10 strains), which is in agreement with the results reported by a previous study [66]. Among all methods, ShoRAH has the lowest performance in terms of *Recall* and *Precision*. As indicated by *Predicted Proportion*, aBayesQR is the most accurate method in terms of estimating the population size although it often misses a strain with the lowest frequency when applied to reconstruction of a quasispecies consisting of 10 strains. ViQuaS and ShoRAH typically overestimates the number of strains especially at low diversity levels. aBayesQR is the best method in terms of *Reconstruction Rate* at all levels of diversity. In terms of frequency estimation, aBayesQR overall outperforms all the other methods whereas PredictHaplo shows the highest *JSD* due to its drawback of underestimating the number of strains. Note that both ViQuaS and ShoRAH exhibit significantly increased (i.e., deteriorated) *JSD* at low diversity levels. This fact, along with the low *Recall* and *Precision* scores they have in low diversity settings, indicates that state-of-the-art methods experience major difficulties when attempting to reconstruct viral quasispecies in those settings, as also observed in [28, 66] and [35].

In the second set of experiments, we study the effects that sequencing

Table 3.2: Performance comparison of different methods for varied error rates (err) on simulated data.

		5 strains					10 strains				
	err(%)	0.1	0.2	0.3	0.4	0.5	0.1	0.2	0.3	0.4	0.5
Recall	aBayesQR	**0.6840**	**0.5920**	**0.5840**	**0.5400**	**0.4840**	**0.6080**	**0.5150**	**0.4620**	**0.4200**	**0.3570**
	ShoRAH	0.1300	0.1420	0.1280	0.1300	0.1060	0.0740	0.0490	0.0570	0.0850	0.0880
	ViQuaS	0.6040	0.5300	0.4740	0.4160	0.3540	0.3730	0.2890	0.2200	0.1960	0.1590
	PredictHaplo	-	-	-	-	-	0.1021	0.1031	0.1072	0.1000	0.1194
Precision	aBayesQR	**0.6826**	**0.5954**	**0.6105**	**0.5565**	**0.4958**	**0.6610**	**0.5852**	**0.5377**	**0.5031**	0.4311
	ShoRAH	0.1240	0.1425	0.1274	0.1260	0.1057	0.0498	0.0318	0.0370	0.0516	0.0547
	ViQuaS	0.4559	0.3505	0.2757	0.2067	0.1453	0.2973	0.1889	0.1437	0.1126	0.0634
	PredictHaplo	-	-	-	-	-	0.4509	0.4287	0.4502	0.4325	**0.4787**
PredProp	aBayesQR	**1.0120**	1.0340	**0.9940**	**1.0000**	**1.0140**	**0.9240**	**0.8930**	**0.8780**	**0.8660**	**0.8650**
	ShoRAH	1.0780	**1.0160**	1.0360	1.0580	1.0520	1.6710	1.7460	1.7690	1.8500	1.7950
	ViQuaS	1.4080	1.6860	1.9700	2.3420	2.8280	1.5450	1.8430	1.9540	2.1550	2.5680
	PredictHaplo	-	-	-	-	-	0.1947	0.2000	0.2000	0.1970	0.2082
ReconRate	aBayesQR	**0.9971**	**0.9963**	**0.9957**	**0.9950**	**0.9937**	**0.9952**	**0.9941**	**0.9937**	**0.9925**	**0.9917**
	ShoRAH	0.9891	0.9884	0.9884	0.9879	0.9867	0.9899	0.9891	0.9891	0.9889	0.9887
	ViQuaS	0.9917	0.9923	0.9912	0.9829	0.9805	0.9899	0.9886	0.9879	0.9865	0.9860
	PredictHaplo	-	-	-	-	-	0.9850	0.9850	0.9848	0.9854	0.9848
JSD	aBayesQR	**0.0008**	**0.0018**	**0.0027**	**0.0028**	0.0045	**0.0023**	**0.0035**	**0.0041**	**0.0055**	**0.0060**
	ShoRAH	0.0047	0.0029	0.0038	0.0066	**0.0041**	0.0422	0.0499	0.0493	0.0522	0.0529
	ViQuaS	0.0222	0.0294	0.0384	0.0487	0.0703	0.0495	0.0574	0.0652	0.0696	0.0862
	PredictHaplo	-	-	-	-	-	0.1971	0.2024	0.1861	0.1973	0.1908

Performance comparison of aBayesQR, ShoRAH, ViQuaS and PredictHaplo in terms of *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD* on the simulated data with $div = 3\%$ and $cov = 500\times$ vs. err for a mixture of 5 and 10 viral strains. Averaged PredictHaplo results are reported if it provides answers for more than 50% of data sets. Boldface value indicate the best performance for each err(%)

errors have on the performance of aBayesQR, ShoRAH, ViQuaS and PredictHaplo. In particular, the sequencing error is varied from 0.1% to 0.5% (specifically, $err \in \{0.1\%, 0.2\%, 0.3\%, 0.4\%, 0.5\%\}$), reflecting the range of errors observed in the current and anticipated in future NSG technologies (e.g., the error rates of Illumina's Miseq have been reported to be below 0.4% in

[62] and as high as 0.49 in [63]). We set *div* to be 3%[3] and set *cov* to be 500× which is the same as in the first experiment in Section 3.3.1.

As seen in Table 3.2, aBayesQR outperforms ShoRAH, ViQuaS and PredictHaplo over the considered range of *err* achieving the best scores overall for all 5 metrics. As expected, the performances of all methods deteriorate as *err* increases. Since PredictHaplo failed to generate results in most of the instances of the reconstruction problem involving a mixture of 5 strains, its results are not reported. For the problem involving a mixture with 10 strains, PredictHaplo did run successfully in most of the instances but significantly underdetermined the number of strains; on average, its *Predicted Proportion* is around 0.2 (while its *Precision*, as argued earlier in the chapter, is somewhat misleadingly good). ViQuaS overestimates the number of strains in all instances; we observe that as *err* increases, ViQuaS generates increasingly more false negative viral strains which adversely affects *Precision* and *JSD*. Even though ShoRAH exhibits the lowest perfect reconstruction scores, it achieves better performance than ViQuaS in terms of frequency estimation and the number of reconstructed strains (i.e., *Predicted Proportion* and *JSD*) when applied to reconstruction of mixtures with 5 strains. For a mixture of 10 strains, however, ShoRAH overestimated the number of strains, which also leads to higher *JSD* scores.

In the last set of experiments, we test the performance of the proposed

[3]This matches typical variations in the HIV *pol* gene which range between 3% and 5% [28].

Table 3.3: Performance comparison of different methods for varied coverages (*cov*) on simulated data.

		5 strains			10 strains		
	cov(×)	250	500	750	250	500	750
Recall	aBayesQR	**0.4440**	**0.5840**	**0.5920**	**0.4450**	**0.4620**	**0.4450**
	ShoRAH	0.1820	0.1280	0.0900	0.0590	0.0570	0.3220
	ViQuaS	0.4220	0.4740	0.4840	0.1970	0.2200	0.2580
	PredictHaplo	-	-	-	0.1070	0.1072	0.1094
Precision	aBayesQR	**0.4231**	**0.6105**	**0.6393**	**0.5131**	**0.5377**	**0.5195**
	ShoRAH	0.1831	0.1274	0.0979	0.0544	0.0370	0.2136
	ViQuaS	0.2291	0.2757	0.3256	0.0965	0.1437	0.1900
	PredictHaplo	-	-	-	0.4558	0.4502	0.4635
PredProp	aBayesQR	1.0920	**0.9940**	**0.9400**	**0.8840**	**0.8780**	**0.8800**
	ShoRAH	**1.0140**	1.0360	1.0660	1.1200	1.7690	1.6440
	ViQuaS	1.9240	1.9700	1.6820	2.2940	1.9540	1.7180
	PredictHaplo	-	-	-	0.2030	0.2000	0.2031
ReconRate	aBayesQR	**0.9941**	**0.9957**	**0.9959**	**0.9939**	**0.9937**	**0.9930**
	ShoRAH	0.9906	0.9884	0.9854	0.9874	0.9891	0.9926
	ViQuaS	0.9905	0.9912	0.9918	0.9861	0.9879	0.9881
	PredictHaplo	-	-	-	0.9854	0.9848	0.9854
JSD	aBayesQR	0.0040	**0.0027**	**0.0026**	**0.0041**	**0.0041**	**0.0049**
	ShoRAH	**0.0021**	0.0038	0.0100	0.0051	0.0493	0.0330
	ViQuaS	0.0454	0.0384	0.0318	0.0839	0.0652	0.0555
	PredictHaplo	-	-	-	0.1946	0.1861	0.1930

Performance comparison of aBayesQR, ShoRAH, ViQuaS and PredictHaplo in terms of *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD* on the simulated data with $div = 3\%$ and $err = 0.3\%$ vs. *cov* for a mixture of 5 and 10 viral strains. Averaged PredictHaplo results are reported if it provides answers for more than 50% of data sets. Boldface value indicate the best performance for each *cov*(×).

algorithm at different coverages $cov \in \{250\times, 500\times, 750\times\}$ while fixing the other parameters – specifically, diversity is set to 3%, which is the same as in the second experiment in Table 3.2, and sequencing error rate is set to 0.3%, which emulates the error rates of Illumina's Miseq ($< 0.4\%$ [62]). Per-

Table 3.4: Running time comparisons (sec).

	5 strains			10 strains		
$cov(\times)$	250	500	750	250	500	750
aBayesQR	96	113	236	451	739	1606
ShoRAH	559	2005	5265	2716	11473	22923
ViQuaS	93	337	718	360	1300	13350
PredictHaplo	-	-	-	93	143	187

Running time comparisons of aBayesQR, ShoRAH, ViQuaS and PredictHaplo on the simulated data with $cov \in \{250\times, 500\times, 750\times\}$, div=3% and err=0.3%, measured on a Linux OS desktop with 3.06GHz CPU and 8Gb RAM (Intel Core i7 880 processor). PredictHaplo results are shown if it provides answers for more than 50% of data sets.

formance of four algorithms as a function of coverage is compared in Table 3.3, demonstrating superiority of aBayesQR in all five metrics of interest.

Runtimes of each of the algorithms applied to this test set as a function of *cov* are shown in Table 3.4; note that this characterization of speed (i.e., complexity vs. *cov*) is the most meaningful one to study since the coverage is a main factor affecting the runtime of performing a reconstruction task. The speed is measured on a Linux OS desktop with 3.06GHz CPU and 8GbRAM (Intel Core i7 880 processor). When it completes the task and provides a solution, PredictHaplo is the most efficient among all schemes; however, this method fails to provide answers in most of instances on a mixture of 5 strains. Among the remaining 3 algorithms, our aBayesQR demonstrates the best time efficiency for $cov \geq 500$ while ShoRAH is the slowest one. ViQuaS is relatively fast at the low coverage $cov = 250$ but its time complexity appears to grow exponentially as *cov* increases, especially in the setting with a mixture of 10

viral strains.

3.3.2 Performance comparison on real HIV data

To further test the performance of our proposed method, we employ it for the analysis of the HIV 5-virus-mix dataset published in [23]. Specifically, we apply our algorithm to reconstruct an *in vitro* generated quasispecies population consisting of 5 known HIV-1 strains: HIV-1_{HXB2}, HIV-$1_{89.6}$, HIV-1_{JR-CSF}, HIV-1_{NL4-3} and HIV-1_{YU2}. Compared to the simulated data set, relative frequencies of the 5 HIV-1 strains are more evenly distributed (about $10\% - 30\%$) and the pairwise distances between strains are higher ($2.61\% - 8.45\%$) [23]. We use the 2×250bp-long paired-end reads provided by Illumina's MiSeq Benchtop Sequencer. The reads are aligned to the HIV-1_{HXB2} reference genome; the reads shorter than 150nt and those having bases with quality scores less than a PHRED threshold of 60 are discarded. We compare the performance of our method applied to gene-wise quasispecies reconstruction of the above described HIV data with that of the competing techniques. Since the current version of ViQuaS software does not support specifying genomic regions, we could not use it in this experiment. When running aBayesQR, we set the parameter η to 0.09 (the setting recommended in Appendix A). Other parameters are set to the same values as the ones used in Section 3.3.1.

We evaluate and report the *Predicted Proportion* (i.e., the fraction of correctly estimated strains as defined in Section 3.3.1) and *Reconstruction Rate*

Table 3.5: Performance comparisons of aBayesQR, ShoRAH and PredictHap on a real HIV-1 5-virus-mix data set.

		p17	p24	p2-p6	PR	RT	RNase	int	vif	vpr	vpu	gp120	gp41	nef
aBayesQR	PredProp	1	1	1	1	1	1	1	1	1.2	1	0.8	0.8	1.2
	$RR(F)_{HXB2}$	100(16.3)	99.4(21.1)	100(22.2)	100(12.5)	98.5(24.3)	100(16.1)	99.9(9.7)	100(9.2)	100(16.4)	99.6(17)	98(30.3)	0(0)	95.8(11.4)
	$RR(F)_{89.6}$	100(27.1)	98.7(17)	100(17.3)	100(17.3)	98.6(18.1)	100(19.7)	100(22.2)	100(20.6)	100(16.3)	92(10.4)	96.5(20.2)	98.9(23.7)	95.5(16.4)
	$RR(F)_{JR-CSF}$	100(31.3)	99.6(24.6)	100(25.8)	100(29.9)	99(21.5)	100(22.1)	100(20.8)	100(32.7)	100(27)	98.8(26.7)	97.7(21.4)	99.1(29.7)	98.2(21.1)
	$RR(F)_{NL4-3}$	100(12.9)	100(21.6)	100(25.6)	100(20.1)	98.9(17.7)	100(30)	100(39.5)	99.8(28.5)	100(23.2)	100(41.3)	96.3(28)	98.8(36.6)	100(31.8)
	$RR(F)_{YU2}$	100(12.4)	99.7(15.8)	100(9.2)	100 (20.3)	99.2 (18.5)	100(12.2)	99.5(7.9)	99.7(9)	100(17.1)	100(4.6)	0(0)	98.6(10.1)	99.2(14)
ShoRAH	PredProp	13	16.4	13.8	8.8	21.8	11.8	13.6	12.8	7.8	4	23.8	19.8	17.4
	$RR(F)_{HXB2}$	100(9.41)	96.7(6.3)	100(6.2)	100(7.8)	98.2(5.8)	100(8.9)	97.5(9.2)	100(5.9)	100(8)	100(8.2)	97.7(15.7)	98.4(4.6)	98.2(14)
	$RR(F)_{89.6}$	100(8.97)	99.7(10.7)	100(15.8)	100(14.5)	98.6(4.6)	100(11.3)	98.9(11.1)	99.8(12.5)	100(6.2)	93.6(14.6)	96.1(9)	98.6(8.4)	98.9(9.7)
	$RR(F)_{JR-CSF}$	100(26.2)	100(13.6)	100(10.7)	100(10.2)	99.8(7.2)	96.4(7.3)	99(7.6)	100(14.5)	100(16.9)	98(20.14)	96.9(9.8)	96.3(6.6)	94.7(4.3)
	$RR(F)_{NL4-3}$	100(9.4)	99.1(5.2)	97.3(7.3)	100(7.8)	98.9(4.4)	99.2(14.8)	99.3(4.8)	99.3(7.3)	100(22.9)	100(20.2)	96.1(4.4)	98.5(4.8)	98.6(1 7.1)
	$RR(F)_{YU2}$	94.2(6.1)	99(5.5)	100(8)	98.3(8.1)	98(7.2)	94.5(9.2)	98.6(6)	95(6.6)	93.2(7.6)	90.8(7.9)	97(4.7)	95.4(7.1)	97.9(5.1)
PredictHaplo	PredProp	1	0.6	1	1	1	0.8	0.8	0.8	1	0.8	0.8	0.8	0.8
	$RR(F)_{HXB2}$	100(17.8)	0(0)	100(18.7)	100(15.2)	100(12.2)	98.9(25.4)	100(12.1)	100(17.7)	100(10.2)	93.17(10.8)	0(0)	0(0)	0(0)
	$RR(F)_{89.6}$	100(19.9)	100(46.4)	100(21.7)	100(22.2)	100(19.4)	100(18.2)	99.8(27.6)	100(20.9)	100(22.1)	0(0)	97.8(20.7)	100(26.7)	98.87(20.7)
	$RR(F)_{JR-CSF}$	100(31.9)	100(21.8)	100(30.3)	100(26.9)	100(23.4)	100(23.2)	100(22.3)	100(24.9)	100(23.7)	100(34.1)	99.7(42.7)	100(28.9)	100(23.2)
	$RR(F)_{NL4-3}$	100(17)	99.1(31.8)	100(16.4)	100(20.9)	100(30.2)	100(33.2)	100(38.1)	100(36.6)	100(35.5)	100(47.1)	100(28.6)	100(32.7)	100(39.3)
	$RR(F)_{YU2}$	100(13.4)	0(0)	100(12.9)	100(14.8)	100(14.7)	0(0)	0(0)	0(0)	100(8.5)	100(7.9)	98.6(7.9)	100(11.7)	100(16.9)

Predicted Proportion (PredProp), *Reconstruction Rate* (RR) and inferred frequencies (F) for aBayesQR, ShoRAH and PredictHaplo applied to reconstruction of HIV-1_{HXB2}, HIV-$1_{89.6}$, HIV-1_{JR-CSF}, HIV-1_{NL4-3} and HIV-1_{YU2} for all 13 genes of the HIV-1 dataset (note: frequencies are reported in parenthesis, both RR and F are expressed in percentages).

in Table 3.5. On this real HIV-1 data set which (as pointed above) has different properties than the simulated data set in Section 3.3.1, aBayesQR is the most accurate among the considered methods in terms of *Predicted Proportion*. PredictHaplo underestimates the population size and reconstructs three or four strains in the 8 considered genes and ShoRAH greatly overestimates the population size for all 13 genes of the HIV-1 data set (e.g., it reconstructs 119 strains in gp120), which is consistent with our simulation results as well as with the results in [66]. aBayesQR and PredictHaplo are tied for the number of genes where all the strains are perfectly reconstructed (5 each); for the remaining genes, PredictHaplo provides a larger number of perfectly reconstructed strains. However, it is worth pointing out that PredictHaplo, designed for identification of HIV haplotypes, missed at least one strain in each of the remaining 8 genes while aBayesQR reconstructed most of the strains on all but two genes, gp120 and gp41. ShoRAH did not perfectly reconstruct any of the 13 genes, which is consistent with the simulation results. Moreover, overestimating the number of strains negatively affects the accuracy of ShoRAH's frequency estimation; for instance, the sum of frequencies corresponding to the most abundant 5 strains does not exceed 50% in 9 out 13 genes (71% is the largest such sum, on *vpu*).

To complement the gene-wise quasispecies reconstruction study with that of a global reconstruction, we consider the HIV-1 *gap-pol* region spanning 4307bp. To efficiently process 355241 paired-end reads that remain after applying a quality filter, we organize the region into a sequence of windows

of length 400bp where the consecutive windows overlap by 150bp and run aBayesQR on those windows. The entire region is assembled by connecting strains in the consecutive windows while testing consistency in the overlapping intervals. The number of strains retrieved in the global reconstruction is decided by majority voting of the number of strains obtained in each window. The frequencies are estimated by counting reads nearest (in terms of Hamming distance) to each of the reconstructed strains. Following this procedure, both aBayesQR and PredictHaplo could reconstruct all 5 HIV-1 strains in the *gap-pol* region correctly, i.e., they both achieved *Reconstruction rate* of 100 for all 5 strains and *Predicted Proportion* of 1. The frequencies estimated by aBayesQR are 15.21%, 19.34%, 25.56%, 27.61% and 12.27% while those estimated by PredictHap are 13.21%, 13.56%, 25.67%, 19.69% and 27.86%. ShoRAH highly overestimated the number of strains and reported *Predicted Proportion* of 41.8; its five most abundant strains estimated are reported to have frequencies 8.51%, 5.04%, 3.41%, 3.24% and 3.09%.

3.4 Summary and Further Work

In this chapter, we presented a novel maximum-likelihood based approximate algorithm for reconstructing viral quasispecies from high-throughput sequencing data. aBayesQR assembles paired-end short reads into longer fragments based on similarity of the read overlaps and the uncertainty level of non-overlapping regions. The probable sets of partially reconstructed strains are inductively searched and a subset of those strains is extended to efficiently

deduce the most likely set of strains in a quasisepcies. Detection of the population size is embedded into the algorithm and is empirically shown to be very accurate; the number of strains is dynamically adjusted based on the reliability of the partially assembled quasispecies in each extension step. Performance of the developed method is tested on both synthetic datasets and a real HIV-1 dataset. In both settings, the new algorithm outperforms existing techniques in terms of accuracy of the quasispecies size estimation, perfect reconstruction of strains, proportion of correct bases in each reconstructed strain and the estimation of their abundance. A particularly high accuracy is observed in estimating the population size (i.e., the number of strains) and their relative abundance. Tests on synthetic datasets demonstrates that aBayesQR is capable of reconstructing quasispecies at low diversity, showing superior performance in those settings compared to state-of-the-art algorithms. Furthermore, the study on a real HIV-1 dataset demonstrates that our proposed algorithm outperforms or has performance comparable to that of the existing methods in the general setting of viral quasispecies reconstruction.

Chapter 4

Viral quasispecies reconstruction via tensor factorization with successive read removal

4.1 Background and Related Works

The sequential Bayesian inference method, aBayesQR, described in Chapter 3 focused on reconstruction of quasispecies characterized by low diversity. While aBayesQR is indeed more accurate than competing methods in low diversity ($\leq 5\%$) settings, reconstructing quasispecies characterized by both low diversity and highly uneven strain frequencies remains a challenge.

In this chapter, we present a reconstruction method that takes a step towards overcoming limitations of existing techniques and is capable of accurately assembling quasispecies characterized by a wide range of strain frequencies.[1] The method, referred to as TenSQR (Tensor factorization with Successive removal for Quasispecies Reconstruction), represents sequencing data by means of a structured sparse binary tensor. Factorization of such objects, both matrices and tensors, was previously used to enable haplotype assembly of diploid [14] and polyploid [31] genomes. Note that the abundances

[1]This work will appear as [Soyoen Ahn, Ziqi Ke and Haris Vikalo. Viral quasispecies reconstruction via tensor factorization with successive removal. Bioinformatics, 2018 ISMB Special Issue, 2018]. The author of this dissertation is the primary contributor.

of reads generated by sequencing diploid or polyploid haplotypes are near uniform – minor variations in those abundances are primarily due to imperfections of the sample preparation and sequencing steps. Matrix and tensor factorization schemes in [14] and [31] make an explicit assumption that the haplotype frequencies are equal. However, this assumption is clearly violated in the QSR problem. It therefore comes as no surprise that when matrix or tensor factorization is directly applied to reconstruct a heterogeneous mixture of sequences characterized by highly uneven frequencies (by means of forming imbalanced clusters, each collecting reads having originated from one sequence), dominant sequences (large clusters) are typically recovered correctly while the rare sequences (small clusters) are often either missed or reconstructed erroneously [16]. To address this concern, TenSQR successively infers strains in a quasispecies by repeatedly performing the following two steps. In the first step, the sparse tensor is factorized and its missing entries are inferred by alternatingly optimizing the factors; this step is completed by identifying and reconstructing the most abundant strain. In the second step, all the reads deemed to have originated from the reconstructed (the most abundant) strain are removed from the dataset and hence from the originally formed tensor; the number of such reads is indicative of the reconstructed strain's frequency. Then, the first step is performed anew on the reduced dataset to reconstruct the 2^{nd} most abundant strain and so on. These two steps are repeated until all the strains are reconstructed. Since the proposed scheme revisits tensor factorization multiple times, computational complexity of that step becomes a concern. To

mitigate it, we exploit the special structure of the problem and propose a novel majority-voting based efficient alternating minimization scheme for sparse binary tensor factorization. We show that the convergence of the successive strain reconstruction and data removal procedure is guaranteed, and that the proposed scheme allows detection of deletions in the reconstructed strains. Finally, the developed framework is augmented by an additional pipeline designed to detect insertions that may be present in some of the reconstructed strains. Our tests on simulated data demonstrate that, unlike the competing methods, the proposed tensor factorization framework for successive strain inference supports reliable discovery and accurate reconstruction of rare strains existing in highly imbalanced populations even when the population diversity is low. In particular, TenSQR compares favorably to state-of-the-art methods at diversities 1-10%, and detects deletions in strains with low abundance. Performance of TenSQR on a real HIV-1 dataset demonstrates TenSQR's ability to reliably reconstruct quasispecies in more general settings. Furthermore, we employ our method to reconstruct full-length strains in a Zika virus sample.

4.2 Method

4.2.1 Problem formulation

Let $Q = \{q_i, i = 1, \cdots, k\}$ denote the set of k viral quasispecies strains that differ from each other at a number of variant sites, and let $R = \{r_j, j = 1, \cdots, m\}$ denote the set of reads generated by sequencing the strains in Q; relative ordering of the reads is determined by aligning them to a reference

genome. Homozygous sites (i.e., the sites containing alleles common to all strains) are not utilized by our tensor model; instead, viral haplotypes are reconstructed using heterozygous sites that have abundance of alleles above a predetermined threshold (i.e., sites that are with high confidence declared to be variants). Note that the homozygous sites are later used to assemble full-length viral strains, as detailed in Section 4.2.3.

Let us organize the data (i.e., information about the variant sites provided by the paired-end reads) in an $m \times n$ read fragment matrix $\mathbf{F}'$, where the rows correspond to reads and columns correspond to variant positions in the sequences. A convenient numerical representation of $\mathbf{F}'$ is obtained by denoting nucleotides with 4-dimensional standard unit vectors $\mathbf{e}_i^{(4)}$, $1 \leq i \leq 4$, with 0s in all positions except the i^{th} one that has value 1 (e.g., $\mathbf{e}_1^{(4)} = [1\ 0\ 0\ 0]$, $\mathbf{e}_2^{(4)} = [0\ 1\ 0\ 0]$ and so on). This leads to a representation of the read fragment matrix $\mathbf{F}'$ by means of a binary tensors $\mathcal{F}$ whose fibers represent nucleotides and horizontal slices correspond to reads. $\mathcal{F}$ can be thought of as being obtained by sparsely sampling an underlying tensor $\mathcal{T}$ whose fibers are standard unit vectors $\mathbf{e}_i^{(4)}$; sampling is potentially erroneous due to sequencing errors. To arrive at a tensor factorization formulation of the problem, it is useful to point out that $\mathcal{T}$ can be thought of as being obtained by multiplying a read membership indicator matrix $\mathbf{M}$ and a binary tensor $\mathcal{H}$ that encodes the true viral haplotype information – namely, fibers of $\mathcal{H}$ are standard unit vectors $\mathbf{e}_i^{(4)}$ representing alleles while each lateral slice of $\mathcal{H}$ is one of the k viral haplotypes that need to be reconstructed. Moreover, the indicator matrix $\mathbf{M}$ has

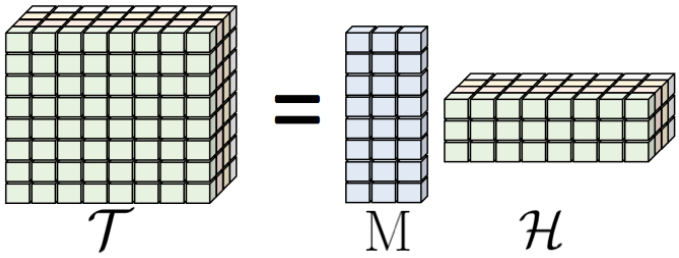

Figure 4.1: *An illustration of the tensor factorization representation of the viral quasispecies assembly problem.*

for rows the standard unit vectors $\mathbf{e}_i^{(k)}$, $1 \leq i \leq k$, with 0s in all positions except the i^{th} one that has value 1. If, for example, the j^{th} row of $\mathbf{M}$ is $\mathbf{e}_l^{(k)}$, then that indicates the j^{th} read was obtained by "sampling" (i.e., via shotgun sequencing) the l^{th} viral haplotype. Figure 4.1 illustrates the representation of $\mathcal{T}$ by means of a product of $\mathbf{M}$ and $\mathcal{H}$.

We formulate the task of reconstructing the set of viral haplotype sequences $\mathcal{H}$ from the observed reads $\mathcal{F}$ as a collection of $k-1$ tensor factorization problems; following each factorization, sequencing reads associated with the most dominant assembled strain are removed from $\mathcal{F}$ and the factorization is performed anew until all the reads remaining in $\mathcal{F}$ are of the same origin (i.e., come from the same viral haplotype). The tensor factorization procedure is formalized in the next section.

4.2.2 Structured tensor factorization using alternating minimization

Let $\mathbf{F} \in \{0,1\}^{m \times 4n}$ and $\mathbf{H} \in \{0,1\}^{4n \times k}$ be the mode-1 unfoldings of tensors $\mathcal{F}$ and $\mathcal{H}$, respectively. The QSR problem can be cast as the optimiza-

tion

$$\min_{\mathbf{M},\mathbf{H}} \frac{1}{2}\|\mathcal{P}_\Omega(\mathbf{F} - \mathbf{M}\mathbf{H}^\top)\|_F^2, \tag{4.1}$$

where Ω denotes the set of informative entries of $\mathbf{F}$ (i.e., positions of the information provided by the set of reads), $\mathcal{P}_\Omega$ is the projection operator (formalizing the sampling of virus strains by reads) and $\|\cdot\|_F$ denotes the Frobenius norm of its argument. This is a computationally challenging optimization problem that can be approximately solved by means of alternating minimization, i.e., alternately solving (4.1) for either $\mathbf{M}$ or $\mathbf{H}$ while keeping the other one fixed [34]. In particular, given the current estimates $\mathbf{M}_t$ and $\mathbf{H}_t$, we alternately update

$$\mathbf{M}_{t+1} = \arg\min_{\mathbf{M}\in\{0,1\}^{m\times k}} \frac{1}{2}\sum \|\mathcal{P}_\Omega(\mathbf{F} - \mathbf{M}\mathbf{H}_t^\top)\|_F^2 \tag{4.2}$$

and

$$\mathbf{H}_{t+1} = \arg\min_{\mathbf{H}\in\{0,1\}^{4n\times k}} \frac{1}{2}\sum \|\mathcal{P}_\Omega(\mathbf{F} - \mathbf{M}_{t+1}\mathbf{H}^\top)\|_F^2 \tag{4.3}$$

until a termination criterion is met. Note that one can impose structural constraints on $\mathbf{M}$ to find it efficiently [14]. In particular, $\mathbf{M}$ can be found by examining for each read all possible haplotype associations and selecting the one that minimizes the number of base changes needed to be consistent with the observed information in $\mathbf{F}$ (i.e., minimizing the so-called minimum error correction (MEC) score, first proposed in [49]). Therefore, the complexity of finding $\mathbf{M}$ given $\mathbf{H}$ is $O(mk)$, where m is the number of reads and k denotes the number of viral haplotypes. Now, optimization (4.3) can be performed via, e.g., gradient descent as previously done in [14, 31]; however, we instead exploit the discrete nature of the problem to solve (4.3) much more efficiently.

In particular, we employ the majority voting rule to form consensus sequences among reads that belong to the same cluster, i.e. originate from the same viral haplotype. While the complexity of finding $\mathbf{H}$ given $\mathbf{M}$ is $O(mn)$ for both the majority voting and gradient descent schemes, the former solves (4.3) directly while the latter only takes a step towards the solution. As a result, the convergence of the proposed alternating minimization scheme that employs majority voting to solve (4.3) is significantly faster than that of the scheme relying on gradient descent (please see Appendix C for numerical results illustrating this point). Moreover, we show that the convergence of the proposed alternating minimization procedure is guaranteed (the proof is provided in Appendix B).

4.2.3 Successive reconstruction of viral sequences

The alternating minimization procedure described in Section 4.2.2 is expected to work well in settings where the abundances of different haplotypes are uniform (i.e., equal) and the ploidy is low [14, 31]. However, when the frequencies of components in the mixture are uneven, the described framework is capable of correctly reconstructing dominant sequences but struggles to assemble sequences having low abundances, as pointed out in [16]. The reason for such behavior is that (4.1) emphasizes accurate recovery of dominant haplotypes which significantly contribute to the overall MEC score while neglecting rare ones whose contribution to the MEC score is relatively small. To address this concern, we propose an iterative scheme where upon per-

forming optimization (4.1), the most abundant viral strain is identified and reconstructed from $\mathbf{H}$; following this reconstruction, all the reads assessed to have originated from the reconstructed strain are removed from the dataset (and thus from $\mathbf{F}$). Optimization (4.1) is repeated on the reduced $\mathbf{F}$ to recover the 2^{nd} most abundant strain and the procedure continues until all strains are reconstructed.

Let m_l be the number of rows having unit vectors $\mathbf{e}_l^{(k)}, 1 \leq l \leq k$, in $\mathbf{M}$. The most dominant haplotype can be identified as the w^{th} lateral slice of $\mathcal{H}$, $\mathcal{H}_{w::}$, satisfying $w = \arg\max_l m_l$. While each row of $\mathbf{M}$ is essentially an indicator of the origin of the corresponding read, membership information obtained via optimization (4.1) could be erroneous when the strains in a mixture have non-uniform frequencies. In fact, reads originating from a rare strain are likely to be assigned to a more abundant one, especially when those two strains are highly similar (i.e., in the low diversity setting). Motivated by this observation, we re-examine the reads in $\mathbf{F}$ to identify those originating from the reconstructed viral haplotype using statistical tests described next.

Assume that the sequencing errors are independent and identically distributed across all variant sites for all reads, and that they happen with probability ε. Let n_i denote the number of informative sites of the i^{th} row in $\mathbf{F}'$, $f'_{i\cdot}$, and let d_i be the number of mismatches between $f'_{i\cdot}$ and the recovered haplotype (i.e., the most abundant haplotype reconstructed in the current iteration) counted over the observed nucleotides of $f'_{i\cdot}$. The probability $p_i(x)$ that x or more sequencing errors occur in the i^{th} fragment is given by the

binomial distribution

$$p_i(x) = \mathrm{P}(X_i \geq x) = \sum_{z=x}^{n_i} \binom{n_i}{z} \varepsilon^z (1-\varepsilon)^{n_i - z}.$$

We first construct a significance test to infer if the aforementioned mismatch has been induced by mutations present in some of the informative sites. In particular, if $p_i(d_i)$ is smaller than or equal to a pre-specified p-value α, we declare that not all of the d_i mismatches are sequencing errors, implying at least one of them is due to mutation, and therefore the i^{th} read remains in $\mathbf{F}$. Otherwise, for a read such that $p_i(d_i) > \alpha$, we further examine its origin by considering the probability p_i that d_i sequencing errors occurs in the i^{th} read,

$$p_i = \binom{n_i}{d_i} \varepsilon^{d_i} (1-\varepsilon)^{n_i - d_i}.$$

The reads which satisfy $p_i > \delta_i$ are assessed to have originated from the most dominant strain and thus eliminated from $\mathbf{F}$. The threshold δ_i is defined as $\delta_i = \prod_{j=1}^{n} p_{X_i[j]}$, where $X_i[j]$ is the nucleotide observed at position j of the i^{th} read in $\mathbf{F}'$ and $p_{X_i[j]}$ denotes the probability of observing nucleotide $X_i[j]$ at position j; the latter probability is obtained as the empirical allele frequency distribution at position j. Note that if the mismatches between a read and the recovered sequence are due to mutations rather than sequencing errors, it is less likely that the read originated from the recovered haplotype and thus a higher threshold is used. To provide strong evidence against the null hypothesis of d_i sequencing errors, we set the p-value α to a small number; in particular, we set $\alpha = 10^{-5}$.

Finally, to reconstruct full-length strains q_i, we reinsert homozygous alleles into the reads and form a consensus sequence for each cluster. Abundances of the reconstructed strains are estimated by counting the number of reads in each cluster.

The performance comparison between TenSQR and the single-pass tensor factorization (i.e., AltHap [31]), an approach that does not employ successive data cancelation) can be found in Appendix C.

4.2.4 Determining the number of strains

The scheme outlined in this section requires that the number of strains (i.e., clusters) k be specified prior to performing tensor factorization and successive cancelation. To determine k, we consider the improvement rate of the MEC score defined as [5]

$$\text{MECimpr}(k) = \frac{\text{MEC}(k) - \text{MEC}(k+1)}{\text{MEC}(k)}. \tag{4.4}$$

Recall that the MEC score counts the minimum number of nucleotides that need to be changed in the observed reads so that the modified reads are consistent with having originated from the reconstructed sequences; smaller MEC score indicates higher accuracy of clustering. As we increase k, the MEC score decreases monotonically; however, once k has reached the actual number of clusters, its further increase leads to only small improvements of the MEC score. Therefore, we approach the problem of detecting the number of strains by identifying k for which the MEC improvement rate (MECimpr)

saturates. To detect this point, we compare the MEC improvement rate with a predefined threshold; while one can search for k by increasing it in steps of 1 until the MEC improvement rate saturates, we, for efficiency, update candidate k by relying on the so-called *half-interval* search. In particular, starting from an initial k_0, the number of clusters is updated as $k_\tau \leftarrow 2k_{\tau-1}$ until $\text{MECimpr}(k_\tau) \leq \eta$; at this point the number of clusters starts to decrease as $k_{\tau+1} \leftarrow \lfloor (k_\tau + \max\{1, k_i\})/2 \rfloor$ where $\{i \in \{1, \cdots, \tau-1\} : k_i \leq k_\tau\}$. Once $\text{MECimpr}(k_\tau) > \eta$, the number of clusters increases again as $k_{\tau+1} \leftarrow \lfloor (k_\tau + \min k_i)/2 \rfloor$ where $\{i \in \{1, \cdots, \tau-1\} : k_i > k_\tau\}$. If $k_\tau = k_{\tau-1}$, the search procedure stops by assigning $k_{\tau+1} \leftarrow k_\tau + 1$ which is our estimate of the number of strains. The recommended choice of the threshold η is discussed in [5] where it has been demonstrated that the performance of estimating the number of strains via MEC improvement rate is robust with respect to the choice of the threshold. The described procedure will find the true number of strains starting from an arbitrary $k0$; the closer $k0$ is to the true number of strains, the fewer iterations will be needed. The proposed TenSQR method is formalized as Algorithm 1.

4.2.5 Estimating insertions

Strains of a quasispecies may also contain insertions as compared to a reference genome. In this section we propose a pipeline to detect and reconstruct insertions by processing paired-end reads that were previously filtered out due to inability to match them confidently with the reference genome;

Algorithm 1: Tensor factorization with successive removal

Input: Set of reads R aligned to the reference genome
Output: Full length quasispecies Q and frequencies of k strains in Q
Preprocessing: From R, get mode-1 unfolding $\mathbf{F}$ of fragment tensor $\mathcal{F}$
Initial $\tau \leftarrow 0$, MECflag $\leftarrow 0$
while $\tau = 0$ or $k_\tau \neq k_{\tau-1}$ **do**
  for $k \in \{k_\tau, k_\tau + 1\}$ **do**
    $Q_k \leftarrow \emptyset$
    $\mathbf{F} \leftarrow$ mode-1 unfolding $\mathbf{F}$ of fragment tensor $\mathcal{F}$
    while $\mathbf{F} \neq \emptyset$ or $k \geq 1$ **do**
      Initialize $\mathbf{H_0} \leftarrow V\Sigma^{\frac{1}{2}}$ where $U\Sigma V^\top = \mathrm{SVD_k}(\mathcal{P}_\Omega(\mathbf{F}))$
      Repeat
        $\mathbf{M}_{t+1} = \arg\min_{\mathbf{M}} \frac{1}{2}\sum \|\mathcal{P}_\Omega(\mathbf{F} - \mathbf{M}\mathbf{H}_t^\top)\|_F^2$
        $\mathbf{H}_{t+1} = \arg\min_{\mathbf{H}} \frac{1}{2}\sum \|\mathcal{P}_\Omega(\mathbf{F} - \mathbf{M}_{t+1}\mathbf{H}^\top)\|_F^2$
      Until termination criterion is met
      Identify $\mathcal{H}_{w::}$ s.t. $w = \arg\max_l m_l$
      Remove f_i s.t $p_i > \delta_i$ or $d_i = 0$ from $\mathbf{F}$
      Reconstruct $Q_k \leftarrow [Q_k; q_h]$ and estimate frequency of q_h
      $k \leftarrow k - 1$
    end while
    Calculate MEC(k)
  end for
  if MECimpr(k_τ) $\leq \eta$ **do**
    $k_{\tau+1} \leftarrow \lfloor (k_\tau + \max\{1, k_i\})/2 \rfloor, \{i \in \{1, \cdots, \tau\text{-}1\} : k_i \leq k_\tau\}$
    MECflag $\leftarrow 1$
  else do
    if MECflag $= 0$ **Do**
      $k_{\tau+1} \leftarrow 2k_\tau$
    else do
      $k_{\tau+1} \leftarrow \lfloor (k_\tau + \min k_i)/2 \rfloor, \{i \in \{1, \cdots, \tau\text{-}1\} : k_i > k_\tau\}$
    end if
  end if
  $\tau \leftarrow \tau + 1$
end while
$k \leftarrow k_\tau + 1$
$Q \leftarrow Q_k$

therefore, those reads have not been used in tensor factorization. The idea is that some of the discarded reads that have high base-calling quality scores might have not been matched successfully to the reference because they originate from a region that was inserted into one of the strains of the quasispecies.

The three major steps of the proposed procedure are the following:

1. Infer the origin of the filtered-out paired-end reads having high base-calling quality scores.

Let us denote two sequences in a paired-end read as (r_h, r_l) where the first one is a read having high mapping score and the other one a read which is not mapped or is only partially aligned to the reference (i.e., has a low mapping score) due to an insertion. In order to infer the origin of the paired-end read (r_h, r_l), we align the mapped read r_h to the position identified by the alignment software and count the number of mismatches between the read r_h and each of k reconstructed strains. The strain q_I having the smallest Hamming distance to the paired-end read is inferred as the origin of the paired-end read (r_h, r_l) and thus contains an insertion.

2. Alignment of r_l

Having identified strain q_I as the origin of the paired-end read (r_h, r_l), we next attempt to find the most likely position of r_l relative to q_I. To this end, we examine subsequences that consist of the first and last l nucleotides of r_l and compare them to every l-nucleotides long subsequence of q_I. Note that we assume $l < L_{r_l}$, where L_{r_l} denotes the length of read r_l. To identify

the best placement of r_l along q_I, we measure alignment scores for $2(L_g - l)$ windows, where L_g is the length of q_I, and choose the one with the largest score. More specifically, the alignment score is updated for each window at each position of $1 \leq i \leq l$; $score_{i+1} \leftarrow score_i + N_{match}$ if the i^{th} nucleotide of the l-nucleotides long subsequences is matched with one in q_I, otherwise the score is updated as $score_{i+1} \leftarrow score_i - (N_{mismatch}+1)^2$ where N_{match} and $N_{mismatch}$ are the number of consecutive matches and mismatches, respectively; the scoring function penalizes consecutive mismatches while favoring consecutive matches to form a reliable estimate of the placement for r_l.

Given a relative position of r_l along q_I, we infer the inserted position (i.e., the position at which r_l starts to differ from q_I due to insertions) by relying on the same scoring function used to determine the best alignment position. In particular, the score is computed for each position from 1 to L_{r_I} for reads whose l-nucleotides-long prefix maps to q_I and from L_{r_I} to i for those whose l-nucleotides-long suffix maps to q_I; then the position at which insertion starts is estimated as $I + 1$, where $I = \arg\max_i score_i$. For computational efficiency, the updates of $score_i$ are terminated once $score_i < 0$.

3. Recover inserted sequences

Given the alignment and the position at which the insertion starts, recovery of the insertion is readily performed by constructing two consensus sequences – one built using the collection of reads whose l-nucleotides-long prefix maps to q_I and the other whose l-nucleotides-long suffix maps to q_I.

Finally, the two consensus sequences are aligned and merged to recover the entire insertion.

Preliminary results demonstrating its performance can be found in Appendix C.

4.3 Results and Discussion

4.3.1 Performance comparison on simulated data

We first test the performance of TenSQR on the synthetic data generated by emulating high-throughput sequencing of quasispecies samples. Viral strains in a quasispecies are generated by introducing independent mutations at uniformly random locations of a randomly generated reference genome of length 1300bp (this is a typical length of a gene in the *pol* region of the HIV-1 genome). 2×250bp-long Illumina's MiSeq reads with inserts that have average length and standard deviation 150bp and 30bp, respectively, uniformly sample the mixture of viral strains at a coverage rate of $500\times$ per strain. The reads are aligned to a reference using BWA-MEM algorithm with the default settings [44]; reads shorter than 100bp or having mapping quality score lower than 40 are filtered out. Simulated data is categorized into 40 different sets, each consisting of 50 samples, according to diversity ($div\%$), sequencing error rate (ε) and the number of strains in a quasispecies (and hence frequencies of the strains). Diversity, defined as the average Hamming distance between different strains in a quasispecies, varies from 1% to 10%. Sequencing error rate is set to $\varepsilon = 7 \times 10^{-3}$ and $\varepsilon = 2 \times 10^{-3}$, the typical error rates in MiSeq

datasets before and after quality trimming with error correction, respectively [65]. For each configuration of parameters we consider two mixture sets, each consisting of 5 and 10 viral strains. Frequencies of strains are chosen to approximately follow geometric distribution so as to emulate uneven populations which include strains with low abundance; relative strain frequencies for the 5-strain mix are $(0.5, 0.3, 0.15, 0.04, 0.01)$ while those for the 10-strain mix are $(0.36, 0.24, 0.16, 0.08, 0.055, 0.04, 0.03, 0.02, 0.01, 0.005)$. Therefore, the total coverage for the 5-strain and 10-strain population are $2500\times$ and $5000\times$, respectively, implying that strains having relative frequencies 0.0023 or higher in the 5-strain case and those with relative frequencies 0.0046 or higher in the 10-strain case are covered by sequencing reads with probability 0.99 [28].

We compare the performance of TenSQR on the generated datasets with publicly available softwares PredictHaplo [57], ShoRAH [78], ViQuaS [35], and aBayesQR [5], in terms of *Recall*, *Precision*, *Predicted Proportion*, *Reconstruction Rate* and *Jensen-Shannon divergence (JSD)*. To assess the ability of the compared methods to reconstruct viral strains perfectly (without errors), *Recall* is defined as the fraction of the reconstructed strains that match the true strains in a quasispecies, i.e., $Recall = \frac{TP}{TP+FN}$ while *Precision* reports the fraction of true sequences among the reconstructed strains, i.e., $Precision = \frac{TP}{TP+FP}$. We further report *Predicted Proportion*, defined as the ratio of the estimated and the true population size, thus measuring accuracy of the methods' population size prediction. Note that the proximity of the value of this metric to 1 indicates accuracy of the population size estimate. To assess the

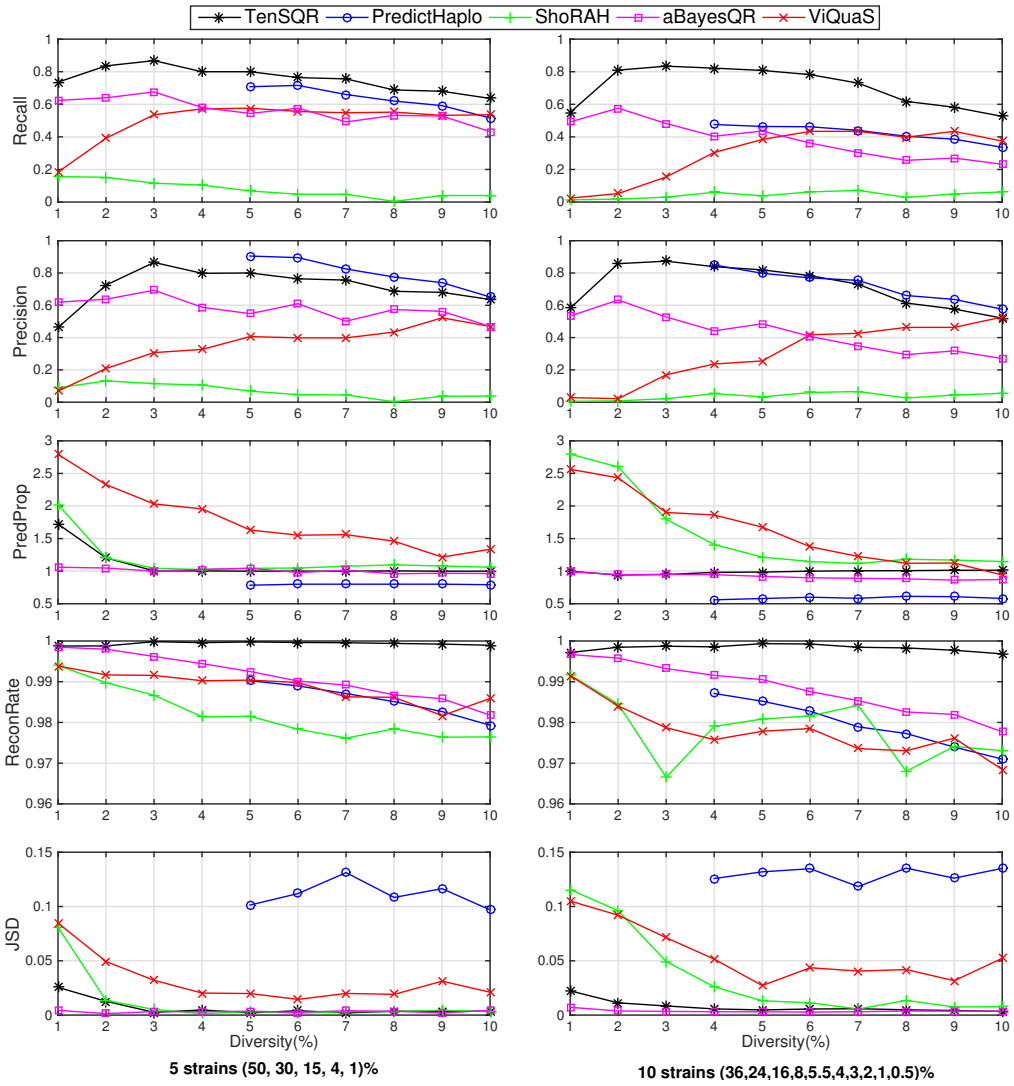

Figure 4.2: Performance comparison of TenSQR, aBayesQR, ShoRAH, ViQuaS and PredictHaplo in terms of *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD* on the simulated data with $\varepsilon = 2 \times 10^{-3}$ for a mixture of (a) 5 viral strains and (b) 10 viral strains. (For the plots that include error bars, please see the corresponding Figure C.2 in Appendix C.)

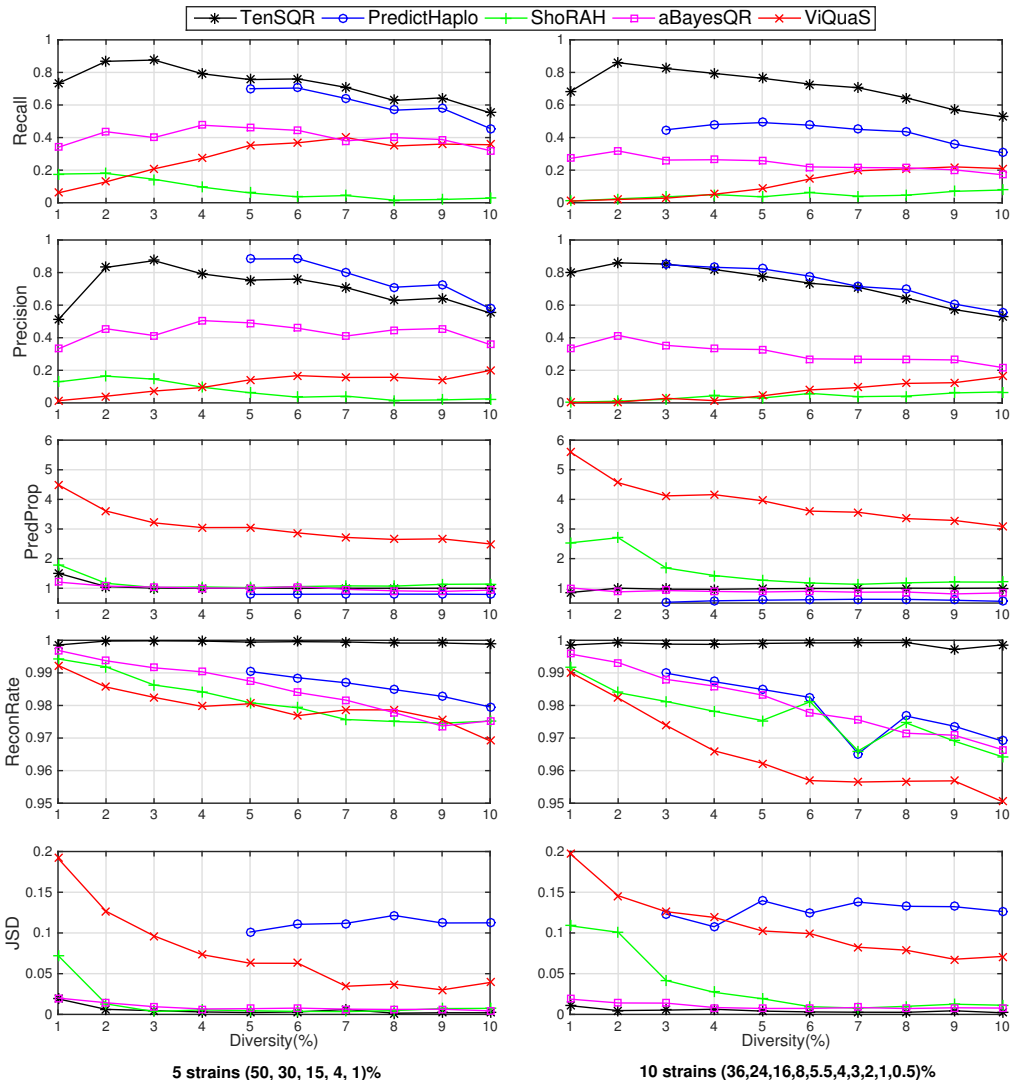

Figure 4.3: Performance comparison of TenSQR, aBayesQR, ShoRAH, ViQuaS and PredictHaplo in terms of *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD* on the simulated data with $\varepsilon = 7 \times 10^{-3}$ for a mixture of (a) 5 viral strains and (b) 10 viral strains. (For the plots that include error bars, please see the corresponding Figure C.3 in Appendix C.)

degree of accuracy of each reconstructed strain, we use

$$\textit{Reconstruction Rate} = \frac{1}{k}\sum_{i=1}^{k}\left(1 - \frac{HD(q_i, \hat{q}_i)}{G}\right),$$

where G denotes the length of a reference genome, k is the number of strains in a quasispecies, and q_i and $\hat{q}_i$ denote the i^{th} true strain and its nearest sequence among the k reconstructed ones, respectively. Finally, *JSD* measures accuracy of the estimated frequencies of the reconstructed strains, i.e., quantifies similarity between two inferred distributions. Formally, *JSD* between a true distribution P and its approximation Q is defined as

$$JSD(P||Q) = \frac{1}{2}D(P||M) + \frac{1}{2}D(Q||M),$$

where $D(\cdot||\cdot)$ denotes Kullback-Leibler (KL) divergence, M is defined as $M = \frac{1}{2}(P+Q)$, and the KL divergence is found as

$$D(P||Q) = \sum_{i=1}^{n} P(i)\log\frac{P(i)}{Q(i)}.$$

Figure 4.2 and Figure 4.3 compares the values of these five metrics computed for each of the considered reconstruction methods; the metrics are evaluated by averaging over 50 instances for each combination of the parameters and error rate $\varepsilon = 2 \times 10^{-4}$ and $\varepsilon = 7 \times 10^{-4}$, respectively. Note that PredictHaplo does not execute on all instances of the generated datasets and hence we report its performance only when all 50 instances are successfully processed. As can be seen in Figure 4.2 and Figure 4.3, TenSQR outperforms the competing schemes. In particular, TenSQR performs the best at all levels of diversity in terms of both *Recall* and *Reconstruction Rate*. In terms

of *Predicted Proportion* and *JSD*, at $div > 2\%$ TenSQR achieves superior or comparable performance to aBayesQR, which is designed to particularly excel at reconstructing low diversity populations. Note that while *Recall* quantifies the fraction of perfectly reconstructed viral strains, purpose of *Reconstruction Rate* is to assess quality of reconstruction when the assembled viral strains are allowed to have errors in some positions. Therefore, the fact TenSQR's *Recall* and *Reconstruction Rate* are close to 1 indicates that the proposed scheme is capable of reconstructing rare sequences (i.e., with low abundance) present in viral mixtures characterized by a wide range of strain frequencies. PredictHaplo underestimates the number of strains and reconstructs only those that have relative frequencies $\geq 15\%$, which explains its high *Precision* at $div \geq 5\%$. ViQuaS overestimates the population size at all levels of diversity, achieving the highest scores in *Predicted Proportion*; note that the only strains used in calculating ViQuaS performance metrics are those that ViQuaS estimated as having frequencies greater than f_{min}, as recommended by [35]. The strains reconstructed by ShoRAH are consistently shorter than the true strains, which appears to be due to the existence of low coverage regions in the synthetic data sets. ShoRAH completes missing sites on the reconstructed strains using bases from the reference genome, which partially explains why ShoRAH underperforms in terms of *Recall*, *Precision* and *Reconstruction Rate*. In conclusion, low *Predicted Proportion* of PredictHaplo and weak performance of other methods in terms of *Recall* and *Reconstruction Rate* indicate that existing techniques experience major difficulties when attempting to detect and reconstruct rare

Table 4.1: Performance of estimating deletion.

		$div\%$									
l_{del}	$f_{min}\%$	1	2	3	4	5	6	7	8	9	10
100	1	0.08(0.80)	0(0.88)	0(1.02)	0(2.16)	0(2.64)	0(2.20)	0(2.52)	0(3.38)	0(4.76)	0(4.08)
	2	0.02(1.18)	0(1.26)	0(1.70)	0(1.42)	0(2.66)	0(1.58)	0(2.78)	0(3.14)	0(2.72)	0(4.22)
	5	0.06(1.64)	0(1.72)	0(2.12)	0(1.94)	0(1.86)	0(2.20)	0(2.22)	0(2.94)	0(3.24)	0(3.70)
	10	0.08(2.18)	0(2.62)	0(2.20)	0(1.90)	0(2.42)	0(1.92)	0(2.00)	0(3.34)	0(3.00)	0(2.50)
200	1	0.16(1.33)	0(1.38)	0(1.20)	0(1.78)	0(2.46)	0(3.02)	0(2.28)	0(4.02)	0(5.56)	0(4.56)
	2	0.14(1.26)	0(1.20)	0(1.94)	0(1.74)	0(1.78)	0(2.34)	0(1.98)	0(3.58)	0(3.94)	0(4.36)
	5	0.14(2.86)	0(1.67)	0(1.90)	0(2.10)	0(2.20)	0(1.96)	0(2.54)	0(3.16)	0(3.48)	0(4.98)
	10	0.16(2.31)	0(2.20)	0(2.06)	0(2.30)	0(1.70)	0(2.10)	0(2.18)	0(2.98)	0(2.54)	0(2.96)
300	1	0.34(1.00)	0(1.62)	0(1.38)	0(2.24)	0(2.04)	0(3.32)	0(2.52)	0(4.56)	0(5.14)	0(5.00)
	2	0.24(2.00)	0.06(1.64)	0(1.78)	0(1.70)	0(2.02)	0(2.24)	0(2.28)	0(3.74)	0(5.08)	0(3.92)
	5	0.30(2.66)	0.02(2.12)	0(1.92)	0.02(1.65)	0(2.04)	0(2.52)	0(2.88)	0(2.84)	0(2.58)	0(3.60)
	10	0.36(2.72)	0(2.60)	0(2.62)	0(2.10)	0(2.28	0(2.64)	0(2.94)	0(2.58)	0(3.10)	0(3.64)

Performance of TenSQR of estimating deletion in terms of *False Negative* rate of detecting deletions and *Deviation* of estimated deletion length (in parenthesis) on the simulated data with $\varepsilon = 2 \times 10^{-3}$ and $cov = 500\times$ for a mixture of 2 strains, depending on diversity(div) and frequency of the low abundant strain (f_{min}) which includes a deletion of length (l_{del}) 100bp, 200bp and 300bp.

strains.

4.3.2 Evaluating identification of deletion

Following the comparison of performance of TenSQR to state-of-the-art methods, we next evaluate how accurate is TenSQR at estimating long deletions. In particular, we investigate TenSQR's ability to detect a fixed-length deletion in a strain over a range of strain frequencies and diversity levels. To this end, we generate sets of quasispecies consisting of two strains where the length of the abundant strain is 1300bp and deletions of sizes $l_{del} = 100, 200,$ and 300bp are placed into the strain of the lower abundance. We generated 40 benchmark sets of reads emulating sequencing of a mixture of two viral strains

with diversity ranging from 1% to 10% and the lower of two frequencies taking values in $\{1, 2, 5, \text{ and } 10\%\}$; parameters of the sequencing platform are as same as those in Section 4.3.1, i.e., the coverage rate is $500\times$ per strain and the sequencing error rate is set to $\varepsilon = 2 \times 10^{-3}$. 50 instances are generated for each of the total 120 data sets. In this study, the performance of detecting deletions is characterized by means of the false negative rate evaluated over 50 instances of the experiment, and the deviations of the estimated length of deletions from the true length calculated by averaging the deviations over the 50 instances. Since the competing QSR methods considered in Section 4.3.1 are unable to detect deletions, we only show the performance of TenSQR. Note that while HaploClique [71] can predict long deletions, the overlap assembly approach recovers many contigs shorter than true strains instead of reconstructing full-length strains, and thus was not included in the benchmarking results. As evident from the results in Table 4.1, TenSQR is capable of detecting long deletions existing in the viral strains whose frequencies are as low as 1%. The performance of detecting long deletions is exceptional when the viral mixture is not particularly characterized by low diversity ($div = 1\%$); in this setting, the performance under the short sequencing reads (2×250) tends to deteriorate as the length of deletions increases.

Remark: The proposed approach to detecting deletions is tested on a mixture of only two strains for the sake of clarity, so as to allow investigation of the interplay between diversity, strain frequency, and deletion length. The method is, however, applicable to more general settings that involve multiple

strains and/or multiple deleted regions (omitted for brevity).

4.3.3 Performance comparison on gene-wise reconstruction of real HIV-1 data

We further test the performance of TenSQR on the real HIV dataset made publicly available by [23]. An *in vitro* generated quasispecies population consists of 5 known HIV-1 strains (HIV-1_{HXB2}, HIV-$1_{89.6}$, HIV-1_{JR-CSF}, HIV-1_{NL4-3} and HIV-1_{YU2}) with pairwise distances between $2.61\%-8.45\%$ and relative frequencies between 10% and 30% [23]. Paired-end sequencing reads of length 2×250bp generated by Illumina's MiSeq Benchtop Sequencer are aligned to the HIV-1_{HXB2} reference genome. Following [23], to ensure reliable results, reads shorter than 150bp and having quality scores of mapping lower than 60 are discarded. We apply TenSQR to gene-wise reconstruction of the HIV population and compare its performance to that of aBayesQR [5] and PredictHaplo [57], shown to be the most competitive softwares in the benchmarking studies in Section 4.3.1. *Predicted Proportion*, defined in Section 4.3.1 as the ratio of the estimated and the true population size, is evaluated by setting the parameter needed to detect the number of strains in a mixture to $\eta = 0.09$, as recommended by [5]. Since the ground truth information specifying the 5 HIV strains is available (http://bmda.cs.unibas.ch/HivHaploTyper/), we evaluate *Reconstruction Rate* for each recovered individual strain, along with the inferred strain frequencies. Note that the strains in the HIV-1 dataset are more evenly distributed than those in the simulated quasispecies in Section 4.3.1. The results reported in Table 4.2 show that TenSQR reconstructs all of the

Table 4.2: Performance comparisons of TenSQR, aBayesQR and PredictHap on a real HIV-1 5-virus-mix data.

		p17	p24	p2-p6	PR	RT	RNase	int	vif	vpr	vpu	gp120	gp41	nef
TenSQR	PredProp	1	1.6	1	1	1.4	1	1	1	1	1.6	2.2	1.2	0.8
	RR_{HXB2}	100(18.7)	98.9(13.1)	100(17.4)	100(9.9)	99.2(12.1)	100(9.2)	100(8.1)	100(9.6)	100(7.2)	92.8(5.9)	96.0(18)	99.0(11.5)	0(0)
	$RR_{89.6}$	100(18.4)	100(19.6)	100(20.1)	100 (17.2)	98.0(13.5)	100(17.2)	100(16.7)	100(25)	100(19.3)	94.0(15)	97.2(10.3)	100(27.8)	95.7(26)
	RR_{JR-CSF}	100(33.8)	100(33)	100(33.6)	100(21.7)	100(20.7)	100(24.6)	100(23.3)	100(20.5)	100(20.3)	100(31.4)	98.3(33.5)	97.7(18.8)	99.8(19)
	RR_{NL4-3}	100(17)	99.3(19.7)	100(17.2)	100(21.4)	99.5(26.7)	100(37.7)	100(41.2)	100(38.4)	100(46.2)	100(38.8)	99.8(9.2)	99.5(23.2)	99.7(42.7)
	RR_{YU2}	100(12.1)	99.3(14.6)	100(7.7)	99.7(29.8)	99.7(14.5)	100(11.4)	100(10.7)	100(6.5)	100(7.1)	100(4.1)	94.9(10.5)	100(10.2)	98.6(12.3)
aBayesQR	PredProp	1	1	1	1	1	1	1	1	1.2	1	0.8	0.8	1.2
	$RR(F)_{HXB2}$	100(16.3)	99.4(21.1)	100(22.2)	100(12.5)	98.5(24.3)	100(16.1)	99.9(9.7)	100(9.2)	100(16.4)	99.6(17)	98(30.3)	0(0)	95.8(11.4)
	$RR(F)_{89.6}$	100(27.1)	98.7(17)	100(17.3)	100(17.3)	98.6(18.1)	100(19.7)	100(22.2)	100(20.6)	100(16.3)	92(10.4)	96.5(20.2)	98.9(23.7)	95.5(16.4)
	$RR(F)_{JR-CSF}$	100(31.3)	99.6(24.6)	100(25.8)	100(29.9)	99(21.5)	100(22.1)	100(20.8)	100(32.7)	100(27)	98.8(26.7)	97.7(21.4)	99.1(29.7)	98.2(21.1)
	$RR(F)_{NL4-3}$	100(12.9)	100(21.6)	100(25.6)	100(20.1)	98.9(17.7)	100(30)	100(39.5)	99.8(28.5)	100(23.2)	100(41.3)	96.3(28)	98.8(36.6)	100(31.8)
	$RR(F)_{YU2}$	100(12.4)	99.7(15.8)	100(9.2)	100 (20.3)	99.2 (18.5)	100(12.2)	99.5(7.9)	99.7(9)	100(17.1)	100(4.6)	0(0)	98.6(10.1)	99.2(14)
PredictHaplo	PredProp	1	0.6	1	1	1	0.8	0.8	0.8	1	0.8	0.8	0.8	0.8
	$RR(F)_{HXB2}$	100(17.8)	0(0)	100(18.7)	100(15.2)	100(12.2)	98.9(25.4)	100(12.1)	100(17.7)	100(10.2)	93.17(10.8)	0(0)	0(0)	0(0)
	$RR(F)_{89.6}$	100(19.9)	100(46.4)	100(21.7)	100(22.2)	100(19.4)	100(18.2)	99.8(27.6)	100(20.9)	100(22.1)	0(0)	97.8(20.7)	100(26.7)	98.87(20.7)
	$RR(F)_{JR-CSF}$	100(31.9)	100(21.8)	100(30.3)	100(26.9)	100(23.4)	100(23.2)	100(22.3)	100(24.9)	100(23.7)	100(34.1)	99.7(42.7)	100(28.9)	100(23.2)
	$RR(F)_{NL4-3}$	100(17)	99.1(31.8)	100(16.4)	100(20.9)	100(30.2)	100(33.2)	100(38.1)	100(36.6)	100(35.5)	100(47.1)	100(28.6)	100(32.7)	100(39.3)
	$RR(F)_{YU2}$	100(13.4)	0(0)	100(12.9)	100(14.8)	100(14.7)	0(0)	0(0)	0(0)	100(8.5)	100(7.9)	98.6(7.9)	100(11.7)	100(16.9)

Predicted Proportion (PredProp) and Reconstruction Rate (RR (%)) for TenSQR, aBayesQR and PredictHaplo applied to reconstruction of HIV-1HXB2, HIV-189.6, HIV-1JR-CSF, HIV-1NL4-3 and HIV-1YU2 for all 13 genes of the HIV-1 dataset. Values in the genes where all the strains are perfectly reconstructed without errors are denoted as boldface. Frequencies are reported in parenthesis.

5 HIV-1 strains correctly in 6 genes while the other considered methods accomplish the same in 5 genes. Consistent with the results in Section 4.3.1, PredictHaplo, designed for identification of HIV haplotypes, underestimates the number of strains by reconstructing three or four strains in the 8 genes.

4.3.4 Assembly of HIV-1 gag-pol genomes

We further use TenSQR to reconstruct the HIV population on the 4036bp long gag-pol region. Reconstruction of longer regions of viral quasispecies requires higher sequencing coverage than that needed for shorter ones. Therefore, for a reliable reconstruction of a viral population spanning long genome region, we fragment the long region into overlapping blocks, perform reconstruction of the blocks independently, and merge the results to retrieve the full region of interest. Specifically, we split the HIV gag-pol region into a set of blocks of length 500bp, where the consecutive blocks overlap by 250bp. We run TenSQR to perform reconstruction of the viral strains in each of the 18 blocks independently, and merge the results in the consecutive blocks while testing consistency of the strains in the overlapping intervals. In particular, if there are mismatches between the reconstructed strains in the overlapping regions of consecutive blocks, we resolve them by performing majority voting using reads that are covering the mismatched positions. The number of strains retrieved by the global reconstruction procedure is decided via majority voting over the number of strains reconstructed in each block. Following this procedure on the 355241 paired-end reads that remain after applying a

quality filter, TenSQR perfectly reconstructed all 5 HIV-1 strains, achieving *Reconstruction Rate* of 100 for all 5 strains and *Predicted Proportion* of 1. Since the pairwise distances between the 5 HIV-1 strains are relatively high, we estimated frequencies of the viral strains by simply counting the number of reads assigned to the same strain according to the Hamming distance between the reads and the reconstructed strains. The resulting frequencies are $15.21\%, 19.34\%, 25.56\%, 27.61\%$ and 12.27%, which is consistent with the result obtained by aBayesQR ([5]).

4.3.5 Assembly of the Zika virus strains

We apply TenSQR to reconstruct full-length genome of an Asian-lineage Zika virus (ZIKV) sample (accession SRR3332513) that is obtained from a rhesus macaque (animal 393422) on the 4^{th} day of infection [25]. 2×300bp reads ($\sim 30000\times$ coverage) are generated from the sample by the Illumina's MiSeq platform and aligned to the ZIKV reference genome (Genbank KU681081.3) of length 10807 bp using the BWA-MEM algorithm with the default settings [44]. The reads shorter than 100bp and those having mapping quality scores smaller than 40 are discarded. For reliable reconstruction of the full-length genome, we follow the strategy outlined in Section 4.3.4; the full region is split into a sequence of blocks of length 2500bp where the consecutive windows overlap by 500bp. We run TenSQR on those blocks and assemble the entire region by connecting reconstructed strains in the consecutive blocks. Relative strain frequencies are estimated using an expectation-maximization algorithm

described in ([28]). Applying the described procedure to 565979 paired-end reads that pass the quality filter, TenSQR reconstructed 2 full-length viral sequences with relative abundances 74% and 26% that diverge from each other by 0.47% within regions between 200 and 5550bp. Among all the competing methods considered in Section 4.3.1, PredictHaplo is the only one that completed reconstruction within 48 hours. PredictHaplo, which typically underestimates the number of strains – especially in quasispecies characterized by low diversity [5] – reconstructed only one strain which matches the dominant strain reconstructed by TenSQR.

4.4 Conclusion

In this chapter we presented a novel tensor factorization based algorithm for the reconstruction of viral quasispecies from high-throughput sequencing data. In particular, sequencing data is represented by a sparse binary tensor and the viral strains in a quasispecies are reconstructed in an iterative manner; at each iteration, the most abundant sequence among those obtained by factorizing the tensor is selected and data originated from the most abundant sequence is removed from the tensor. Benchmarking tests on synthetic datasets demonstrate that the proposed scheme, referred to as TenSQR, is capable of reconstructing quasispecies characterized by imbalanced frequencies of strains, detecting and recovering low abundant strains more reliably than state-of-the-art algorithms. Further studies on a real HIV-1 and Zika dataset demonstrate that TenSQR outperforms existing methods in more

general settings and is applicable to quasispecies reconstruction from virus-infected patient samples.

Chapter 5

Conclusion and future directions

This dissertation seeks to develop algorithms and provide analysis for two problems that have significant impact on medical genomics: haplotype assembly and viral quasispecies reconstruction.

For the haplotype assembly problem, we developed a deterministic sequential Monte Carlo (i.e., particle filtering) algorithm, ParticleHap, which sequentially infers the haplotype sequence, one SNP site at a time, by exhaustively searching for the most likely extension of the partially assembled haplotype in each step, examining both the possible genotypes and phase. We have tested the performance of the developed algorithm on both 1000 Genomes Project data and synthetic datasets, showing its high accuracy and efficiency over a wide range of the haplotype assembly problem parameters. The algorithm performance has been discussed.

The main goal of ParticleHap is accurate haplotype assembly rather than genotype calling. However, methods that can improve accuracy of genotype calling can be incorporated in the proposed algorithm. For example, the prior information about allele and genotype frequencies or linkage disequilibrium patterns can be incorporated in the proposed algorithm, which may

further improve the accuracy of genotype calling and thus of haplotype assembly. In conclusion, the proposed method provides a framework for joint genotyping and haplotyping that leads to accurate haplotype assembly.

For the viral quasispecies reconstruction problem, we have developed two algorithms that overcomes limitations of existing methods and reliably reconstructs quasispecies with a broad range of compositions, including those characterized by low diversity as well as uneven frequencies of its components. The first algorithm, aBayesQR, is a maximum-likelihood based approximate algorithm designed to reconstruct a mixture of closely related viral strains while the second algorithm, TenSQR, is a successive tensor factorization based algorithm focusing on recovery of rare strains in a viral population and detection of long deletions in those rare strains. Performance of the developed methods has been tested on both synthetic datasets emulating a wide range of experimental parameters and a real HIV-1 dataset, demonstrating the capability of reconstructing quasispecies at low diversity and at a wide range of frequencies of strains.

The presented quasispecies methods can be extended and applied to the problem of estimating the population size and the degree of variation among the constituent species in related fields such as immunogenetics. On a related note, bacterial populations are characterized by having relatively lower mutation rates than viral and thus typically have fewer segregating sites on the sequences in a population. The ability of our methods to perform highly accurate reconstruction in such settings should be further investigated.

The possible direction of future research will also include the development of an improved methodology that permits accurate recovery of long insertions potentially present in rare viral strains.

Softwares implemented for each algorithm are available as follows:
ParticleHap: https://sites.google.com/site/asynoeun/particlehap.
aBayesQR: https://github.com/SoyeonA/aBayesQR.
TenSQR: https://github.com/SoYeonA/TenSQR.

Appendices

Appendix A

Guideline for choosing parameter η

A.1 Guideline for choosing parameter η

Here we discuss the choice of parameter η, a threshold used for assessing value of the metric $MECimpr$ in the process of determining the number of sequences K in a quasispecies. Let $Q = \{q_k, k = 1, \cdots, K\}$ denote a viral population consisting of K strains. The set of reads of length L, $R = \{r_i, i = 1, \cdots, |R|\}$, is generated by a sequencing platform having error rate ϵ and aligned to the reference genome of length G. The MEC score characterizing accuracy of the assembly of strains in Q from reads in R is calculated as

$$\text{MEC}(K) = \sum_{m=1}^{M} \min_{k \in \{1,\cdots,K\}} \sum_{j=1}^{|R|} HD(r_i, q_k).$$

Assume that the sequencing errors are independent and identically distributed across all reads. When the quasispecies recovery is perfect, i.e., when all of the K strains are reconstructed correctly and the relative frequencies of strains are estimated accurately, the MEC score is $|R|L\epsilon$. If a strain with the smallest frequency (f_{min}) is not recognized as a distinct mixture component while the other $K - 1$ sequences are correctly reconstructed, the incorrect clustering of $|R|f_{min}$ reads generated from the rarest quasispecies component induces an extra contribution to the MEC score. The MEC score obtained following

perfect reconstruction of $K-1$ strains and misclassification of the K^{th} strain is

$$\text{MEC}(K-1) = |R|L\epsilon + |R|Lf_{min}d(1-\frac{15}{16}\epsilon),$$

where d is the average diversity rate. The MEC improvement rate achieved by increasing the number of viral strains from $K-1$ to K (and thus reducing the MEC score thanks to a correct clustering of the rarest strain) is

$$\begin{aligned} MECimpr(K-1) &= \frac{\text{MEC}(K-1) - \text{MEC}(K)}{\text{MEC}(K-1)} \\ &= \frac{f_{min}d(1-\frac{15}{16}(1-\epsilon))}{f_{min}d(1-\frac{15}{16}(1-\epsilon))+\epsilon} \\ &\equiv \eta_1. \end{aligned}$$

While η_1 is a potential choice for the threshold, it is beneficial to soften it and allow that $MECimpr(\cdot)$ takes on values slightly below it. To this end, let us also consider the scenario where in addition to the perfect recovery of K strains, an extra strain is erroneously inferred by misclassifying reads that in fact should have been placed in the cluster associated with the most abundant strain (i.e., the one having frequency f_{max}). This reduces evaluated MEC score to an unrealistically low value given by

$$\text{MEC}(K+1) = |R|L\epsilon - |R|Lf_{max}\frac{\epsilon^2}{3}.$$

Improvement of the MEC score due to having an extra (unnecessary) cluster can be expressed as

$$\begin{aligned} MECimpr(K) &= \frac{\text{MEC}(K) - \text{MEC}(K+1)}{\text{MEC}(K)} \\ &= \frac{\epsilon}{3} f_{max} \\ &\equiv \eta_2 \end{aligned}$$

We note that for typical parameter values, $\eta 1 \gg \eta 2$; we choose the threshold η by taking a weighted geometric mean of η_1 and η_2,

$$\eta = (\eta_1^{w_1} \eta_2^{w_2})^{\frac{1}{w_1 + w_2}}.$$

To avoid overestimation of the number of strains, we choose w_1 to be larger than w_2. In our experiments, the ratio $r = w_1/w_2$ was set to 5. We find that the results are fairly robust with respect to the choice of parameter η as demonstrated in Table A.1.

Table A.1: Performances comparison of aBayesQR with different parameter η for varied diversities *div* on simulated data.

		5 strains					10 strains				
	div(%)	1	2	3	4	5	1	2	3	4	5
Recall	0.8η	0.7020	0.7080	0.6840	0.6560	0.6320	0.5800	0.6390	0.6060	0.5810	0.5550
	0.9η	0.7060	0.7120	0.6840	0.6560	0.6300	0.5800	0.6390	0.6080	0.5850	0.5550
	η	0.7080	0.7120	0.6840	0.6560	0.6320	0.5810	0.6380	0.6080	0.5860	0.5550
	1.1η	0.7080	0.7120	0.6840	0.6560	0.6300	0.5780	0.6390	0.6100	0.5850	0.5580
	1.2η	0.7060	0.7120	0.6840	0.6560	0.6340	0.5780	0.6370	0.6100	0.5850	0.5590
Precision	0.8η	0.7019	0.7069	0.6811	0.6397	0.6265	0.6186	0.6874	0.6553	0.6273	0.6094
	0.9η	0.7083	0.7130	0.6811	0.6427	0.6301	0.6190	0.6892	0.6602	0.6325	0.6110
	η	0.7113	0.7130	0.6826	0.6447	0.6319	0.6210	0.6881	0.6610	0.6373	0.6140
	1.1η	0.7113	0.7144	0.6826	0.6470	0.6301	0.6177	0.6892	0.6637	0.6379	0.6238
	1.2η	0.7089	0.7144	0.6841	0.6448	0.6410	0.6181	0.6889	0.6660	0.6398	0.6265
PredProp	0.8η	1.0260	1.0160	1.0140	1.0400	1.0240	0.9720	0.9470	0.9300	0.9310	0.9180
	0.9η	1.0200	1.0120	1.0140	1.0380	1.0160	0.9710	0.9440	0.9250	0.9290	0.9140
	η	1.0180	1.0120	1.0120	1.0360	1.0140	0.9680	0.9440	0.9240	0.9240	0.9100
	1.1η	1.0180	1.0100	1.0120	1.0320	1.0160	0.9690	0.9440	0.9230	0.9210	0.9020
	1.2η	1.0200	1.0100	1.0100	1.0340	1.0060	0.9680	0.9410	0.9200	0.9180	0.8990
ReconRate	0.8η	0.9990	0.9980	0.9971	0.9962	0.9954	0.9975	0.9967	0.9952	0.9941	0.9924
	0.9η	0.9990	0.9982	0.9971	0.9962	0.9953	0.9975	0.9967	0.9952	0.9943	0.9924
	η	0.9990	0.9982	0.9971	0.9961	0.9953	0.9975	0.9967	0.9952	0.9942	0.9924
	1.1η	0.9990	0.9982	0.9971	0.9961	0.9953	0.5951	0.6621	0.6350	0.6094	0.5879
	1.2η	0.9990	0.9982	0.9971	0.9961	0.9951	0.9975	0.9967	0.9952	0.9942	0.9922
JSD	0.8η	0.0022	0.0016	0.0009	0.0010	0.0007	0.0043	0.0025	0.0021	0.0021	0.0023
	0.9η	0.0022	0.0008	0.0009	0.0014	0.0008	0.0043	0.0026	0.0022	0.0021	0.0024
	η	0.0022	0.0008	0.0008	0.0014	0.0008	0.0043	0.0026	0.0023	0.0023	0.0025
	1.1η	0.0022	0.0008	0.0008	0.0014	0.0008	0.0043	0.0026	0.0023	0.0023	0.0030
	1.2η	0.0022	0.0008	0.0008	0.0015	0.0010	0.0043	0.0026	0.0024	0.0024	0.0030

Performances of aBayesQR as a function of diversity with parameter η varied around the recommended value from -20% to $+20\%$; shown are *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD*. The data is generated synthetically, relevant parameters are $err = 0.1\%$ and $cov = 500\times$, simulated is a mixture of 5 and 10 viral strains.

Appendix B

Proof of Convergence

B.1 Proof of Convergence of the Alternating Minimization with Majority Voting

Recall the objective function of the optimization problem,

$$f(\mathbf{M}, \mathbf{H}) = \min_{\mathbf{M},\mathbf{H}} \frac{1}{2} \|\mathcal{P}_\Omega(\mathbf{F} - \mathbf{M}\mathbf{H}^\top)\|_F^2.$$

Given the current estimate $\mathbf{M}_t$ and $\mathbf{H}_t$, we alternately update

$$\mathbf{M}_{t+1} = \arg \min_{\mathbf{M} \in \{0,1\}^{m \times k}} \frac{1}{2} \sum \|\mathcal{P}_\Omega(\mathbf{F} - \mathbf{M}\mathbf{H}_t^\top)\|_F^2$$

and

$$\mathbf{H}_{t+1} = \arg \min_{\mathbf{H} \in \{0,1\}^{4n \times k}} \frac{1}{2} \sum \|\mathcal{P}_\Omega(\mathbf{F} - \mathbf{M}_{t+1}\mathbf{H}^\top)\|_F^2.$$

Having fixed $\mathbf{H}_t$, $\mathbf{M}_{t+1}$ is updated by examining for each read all the possible viral haplotype associations and selecting the one that minimizes the number of base changes needed to make reads consistent with the observed information in $\mathbf{F}$, which implies that

$$f(\mathbf{M}_{t+1}, \mathbf{H}_t) - f(\mathbf{M}_t, \mathbf{H}_t) \leq 0. \tag{B.1}$$

Now let us divide $\mathbf{F}$ into k sub-matrices $\mathbf{F}^i$, $1 \leq i \leq k$, each representing the collection of reads assigned to the i^{th} haplotype in $\mathbf{M}_{t+1}$, and rewrite $f(\mathbf{M}_{t+1}, \mathbf{H}_t)$ as

$$f(\mathbf{M}_{t+1}, \mathbf{H}_t) = \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{n} \|\mathcal{P}_{\Omega_{\mathbf{F}^i(\cdot,j)}}(\mathbf{F}^i(\cdot, j) - \mathbf{M}^i_{t+1}\mathbf{H}_t(j,\cdot)^\top)\|_F^2,$$

where $\mathbf{F}^{(i)}(\cdot, j)$ denotes the j^{th} column of $\mathbf{F}^{(i)}$, $\mathbf{M}^i_{t+1}$ has for rows the standard unit vectors $\mathbf{e}_i$, and $\mathbf{H}_t(j,\cdot)$ is the j^{th} row of $\mathbf{H}_t$ at the t^{th} iteration. Since $\mathbf{H}_{t+1}$ is updated by forming consensus sequences using reads in each sub-matrix clustered by $\mathbf{M}_{t+1}$, we obtain

$$\begin{aligned}
&f(\mathbf{M}_{t+1}, \mathbf{H}_{t+1}) - f(\mathbf{M}_{t+1}, \mathbf{H}_t) \\
&= \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{n} \|\mathcal{P}_{\Omega_{\mathbf{F}^i(\cdot,j)}}(\mathbf{F}^i(\cdot, j) - \mathbf{M}^i_{t+1}\mathbf{H}_{t+1}(j,\cdot)^\top)\|_F^2 \\
&\quad - \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{n} \|\mathcal{P}_{\Omega_{\mathbf{F}^i(\cdot,j)}}(\mathbf{F}^i(\cdot, j) - \mathbf{M}^i_{t+1}\mathbf{H}_t(j,\cdot)^\top)\|_F^2 \\
&= \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{n}((N - N_{\mathbf{H}_{t+1}(j,i)}) - (N - N_{\mathbf{H}_t(j,i)})) \\
&= \frac{1}{2}\sum_{i=1}^{k}\sum_{j=1}^{n}(N_{\mathbf{H}_t(j,i)} - N_{\mathbf{H}_{t+1}(j,i)}) \leq 0
\end{aligned} \tag{B.2}$$

where N denotes the total number of the observed nucleotides in $\mathbf{F}^i(\cdot, j)$ and $N_{\mathbf{H}_t^{(j,i)}}$ is the number of nucleotides corresponding to $\mathbf{H}_t^{(j,i)}$. The combination of expressions (B.1) and (B.1) shows that the proposed alternating minimization procedure is guaranteed to converge.

Appendix C

Additional results from Chapter 4

C.1 Comparing speed of tensor factorization that relies on majority voting with the one that relies on gradient descent

Table C.1: Runtime comparison of majority voting and gradient descent.

		div%									
k	method	1	2	3	4	5	6	7	8	9	10
5	MV	1.00	1.50	1.83	2.64	3.63	3.51	4.17	4.49	4.59	5.95
	GD	31.84	42.40	39.60	78.60	66.04	105.55	93.87	112.49	116.83	202.12
10	MV	3.93	8.75	13.13	19.62	24.96	25.03	28.73	33.51	38.32	36.20
	GD	233.88	321.14	320.79	374.16	426.62	535.04	513.29	752.01	604.85	678.27

Running time comparisons (sec) of majority voting (MV) and gradient descent (GD) for tensor factorization on the simulated data with $\varepsilon = 2 \times 10^{-4}$ and $cov = 500\times$ for a mixture of 5 and 10 strains vs diversity (div), measured on a Linux OS servers with Intel Xeon Phi 7250 (1.4GHz) and 96GB RAM.

We compare runtimes of majority voting and gradient descent applied to solving equation (3) in Section 4.2.2 of the main paper. The dataset is the same as the one used in Section 3.1: mixtures of 5 and 10 strains are sequenced with error rate $\varepsilon = 2 \times 10^{-4}$ and coverage $cov = 500\times$. The speed is measured on a Linux OS server with Intel Xeon Phi 7250 (1.4GHz) and 96GB RAM. As evident from the Table C.1, tensor factorization implemented with majority voting is much more time-efficient than the gradient-descent based approach

for all strain diversity levels.

C.2 Comparing accuracy of viral quasispecies reconstruction based on single-pass tensor factorization (AltHap) with the one that employs multiple tensor factorizations and read removal (TenSQR)

We compare the performance of TenSQR and AltHap [31] on the simulated dataset consisting of 5 viral strains whose synthesis is described in Section 4.3.1. We measure the performances of two schemes in terms of *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD*. Since AltHap requires the number of haplotypes a priori, we do not report *Predicted Proportion*. As shown in Figure C.1, TenSQR significantly outperforms AltHap. This, of course, is expected: AltHap is designed for reconstruction of sequence communities characterized by uniform abundances – an assumption violated in viral quasispecies.

C.3 Performance of recovering insertions

To study insertions recovery, we generated data that emulate the same sequencing platform considered in Sections 4.3.1 and Section 4.3.2 of the paper. Sets of quasispecies consisting of two strains where the length of the abundant strain is 1300bp while the other strain (present at frequencies f_{min} of 5 and 10%) includes insertions of length $l_{ins} = 100$ and 200bp. 50 instances are generated for each of 40 benchmark sets of sequencing reads. Strains in

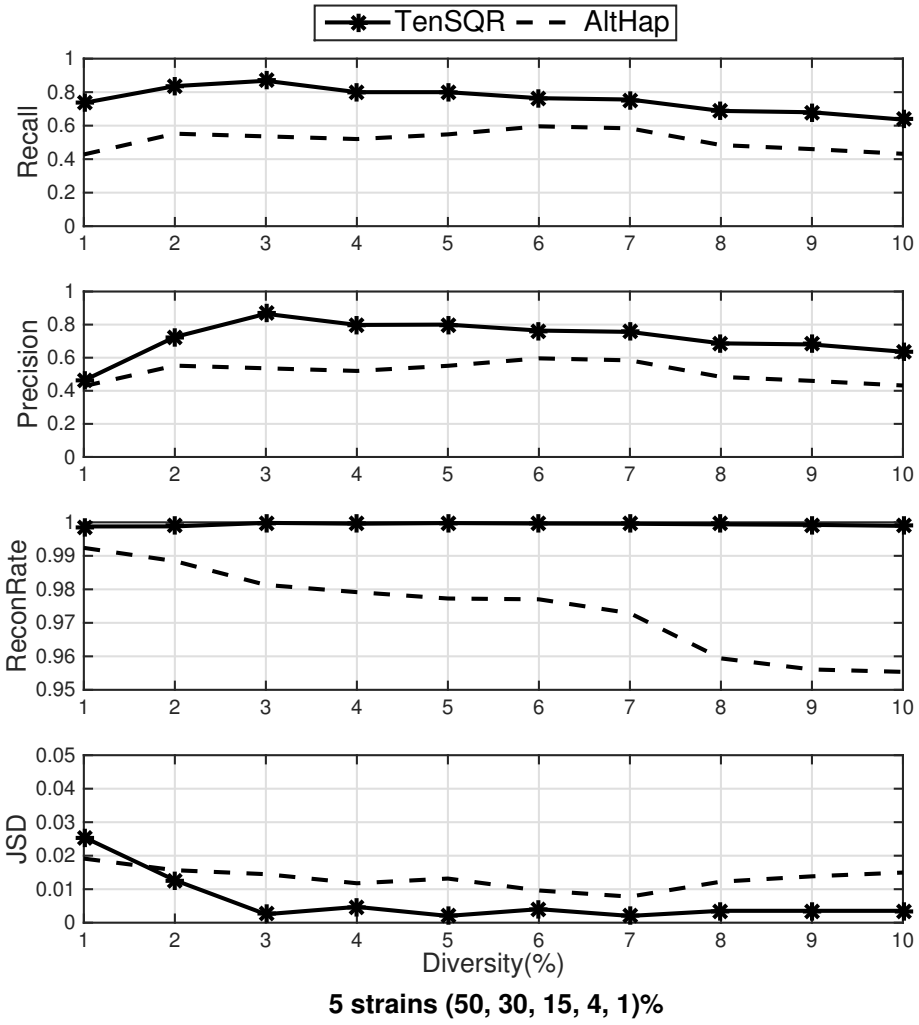

Figure C.1: Performance comparison of TenSQR and AltHap in terms of *Recall*, *Precision*, *Reconstruction Rate* (*ReconRate*) and *JSD* on the simulated data with $\varepsilon = 2 \times 10^{-4}$ for a mixture of 5 viral strains.

the quasispecies have diversity that varies from 1% to 10%; sequencing is performed at the coverage of 500× per strain and is affected by sequencing errors

Table C.2: Performance of recovering insertions.

		$div\%$									
l_{ins}	$f_{min}\%$	1	2	3	4	5	6	7	8	9	10
100	5	0.28	0.12	0.08	0.06	0.12	0.10	0.02	0.10	0.04	0.18
		(0.95)	(0.98)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)
	10	0.14	0.08	0.10	0.04	0.04	0.04	0.04	0.02	0.02	0
		(0.98)	(0.99)	(0.99)	(0.97)	(0.98)	(0.98)	(0.99)	(0.99)	(0.98)	(0.99)
200	5	0.18	0.20	0.14	0.06	0.12	0.06	0.12	0.04	0.14	0.08
		(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)
	10	0.12	0.10	0.02	0.06	0	0.06	0.04	0.02	0.02	0
		(1)	(1)	(0.99)	(1)	(1)	(0.99)	(0.99)	(0.99)	(0.99)	(0.99)

Performance of insertions recovery in terms of the ratio of *False Negative* and *Reconstruction Rate* of insertion (in parentheses) averaged over 50 instances on the simulated dataset with $\varepsilon = 2 \times 10^{-4}$ and $cov = 500\times$ for a mixture of 2 strains, as a function of the diversity(div) and the frequency of the low abundant strain (f_{min}) which contains an insertion of the length (l_{ins}) 100bp and 200bp.

incurring with probability $\varepsilon = 2 \times 10^{-4}$. Performance of insertion detection is evaluated by means of the false negative rate of the detected insertions and the reconstruction rate of the inserted sequences. Table C.2 reports the results of recovering insertions in viral strains. As indicated by the reconstructed rate shown in parentheses, once insertions in the strains are detected, our approach is able to accurately reconstruct the inserted sequences.

C.4 Additional results including error bars

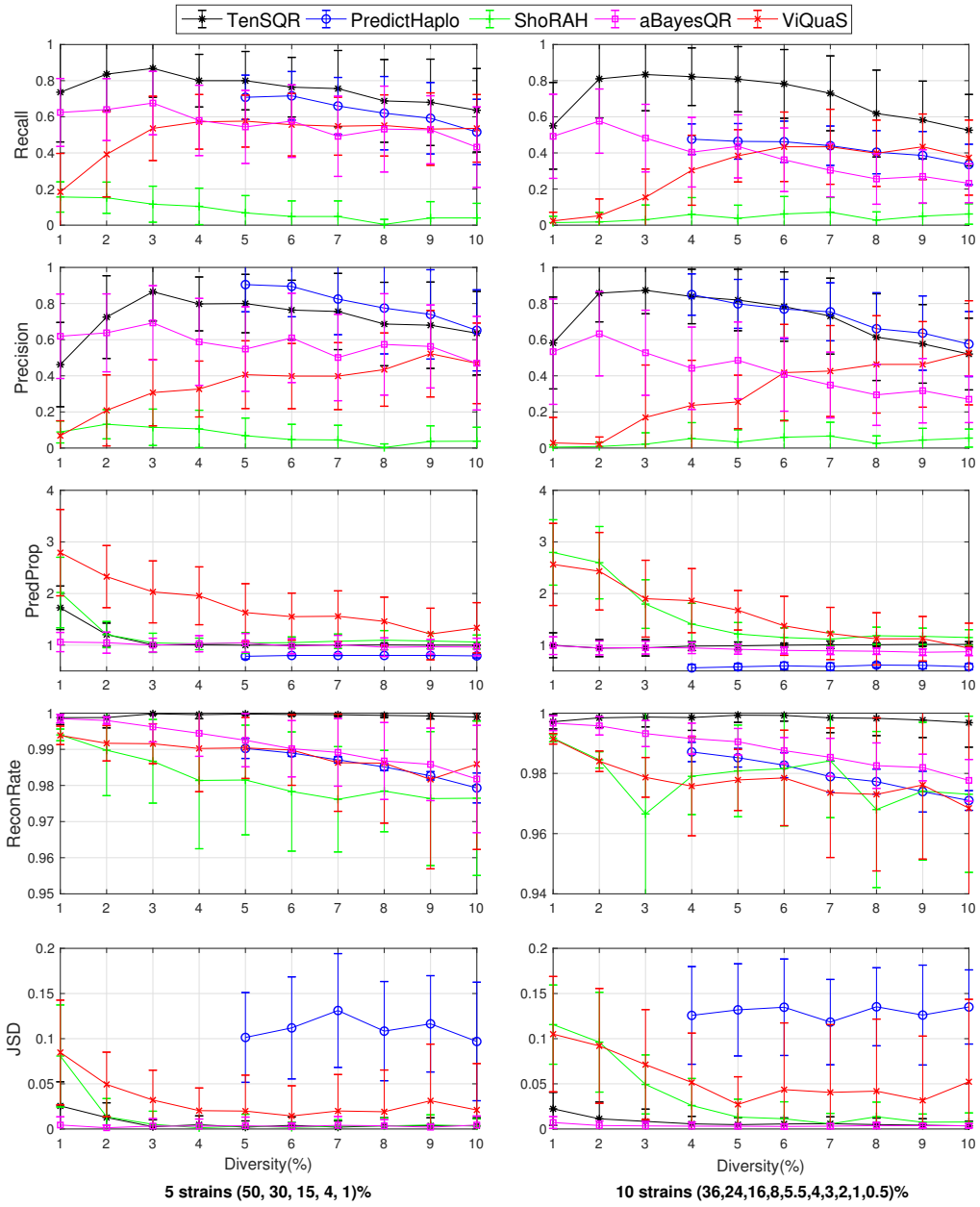

Figure C.2: Performance comparison of TenSQR, aBayesQR, ShoRAH, ViQuaS and PredictHaplo in terms of *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD* on the simulated data with $\varepsilon = 2 \times 10^{-3}$ for a mixture of (a) 5 viral strains and (b) 10 viral strains.

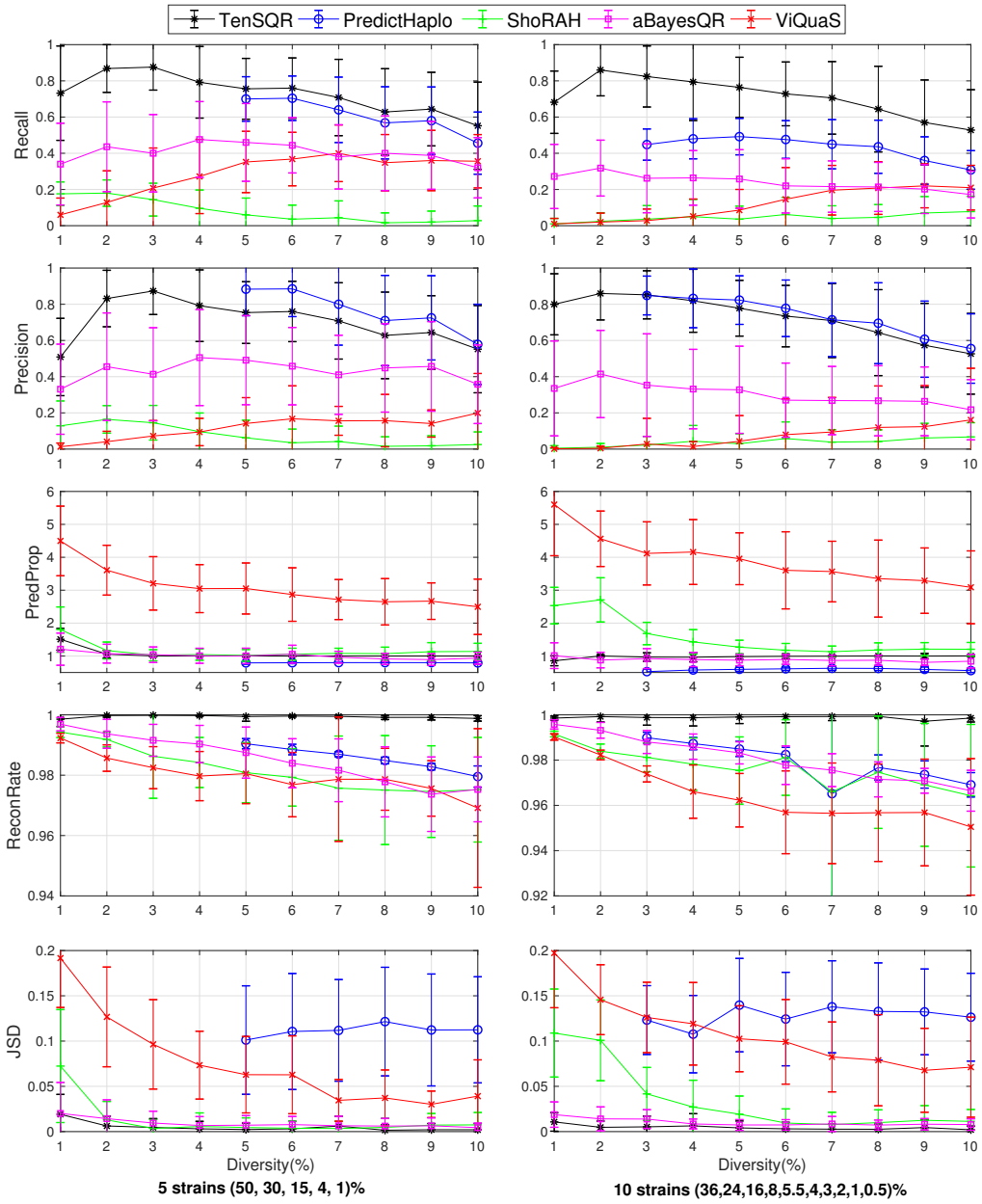

Figure C.3: Performance comparison of TenSQR, aBayesQR, ShoRAH, ViQuaS and PredictHaplo in terms of *Recall*, *Precision*, *Predicted Proportion* (*PredProp*), *Reconstruction Rate* (*ReconRate*) and *JSD* on the simulated data with $\varepsilon = 7 \times 10^{-3}$ for a mixture of (a) 5 viral strains and (b) 10 viral strains.

Bibliography

[1] M Sozio A Panconesi. Fast hare: A fast heuristic for single individual snp haplotype reconstruction. *Algorithms in Bioinformatics*, pages 266–277, 2004.

[2] Derek Aguiar and Sorin Istrail. Hapcompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data. *J Comput Biol*, 19(6):577–90, Jun 2012.

[3] Derek Aguiar and Sorin Istrail. Haplotype assembly in polyploid genomes and identical by descent shared tracts. *Bioinformatics*, 29(13):i352–60, Jul 2013.

[4] Soyeon Ahn and Haris Vikalo. Joint haplotype assembly and genotype calling via sequential monte carlo algorithm. *BMC bioinformatics*, 16(1):223, 2015.

[5] Soyeon Ahn and Haris Vikalo. abayesqr: A bayesian method for reconstruction of viral populations characterized by low diversity. In *International Conference on Research in Computational Molecular Biology*, pages 353–369. Springer, 2017.

[6] John Archer, Greg Baillie, Simon J Watson, Paul Kellam, Andrew Rambaut, and David L Robertson. Analysis of high-depth sequence data

for studying viral diversity: a comparison of next generation sequencing platforms using segminator ii. *BMC bioinformatics*, 13(1):47, 2012.

[7] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *EEE Transaction on signal processing*, 50:174 – 188, 2002.

[8] Irina Astrovskaya, Bassam Tork, Serghei Mangul, Kelly Westbrooks, Ion Măndoiu, Peter Balfe, and Alex Zelikovsky. Inferring viral quasispecies spectra from 454 pyrosequencing reads. *BMC bioinformatics*, 12(6):1, 2011.

[9] Vikas Bansal and Vineet Bafna. Hapcut: an efficient and accurate algorithm for the haplotype assembly problem. *Bioinformatics*, 24(16):i153–9, Aug 2008.

[10] Vikas Bansal, Aaron L Halpern, Nelson Axelrod, and Vineet Bafna. An mcmc algorithm for haplotype assembly from whole-genome sequence data. *Genome Res*, 18(8):1336–46, Aug 2008.

[11] Markus Bauer, Gunnar W Klau, and Knut Reinert. Accurate multiple sequence-structure alignment of RNA sequences using combinatorial optimization. *BMC Bioinformatics*, 8:271, 2007.

[12] MS Bayzid, MM Alam, A Mueen, and MS Rahman. Hmec: A heuristic algorithm for individual haplotyping with minimum error correction. *ISRN Bioinformatics*, 2013.

[13] Niko Beerenwinkel, Huldrych F Günthard, Volker Roth, and Karin J Metzner. Challenges and opportunities in estimating viral genetic diversity from next-generation sequencing data. *Frontiers in microbiology*, 3, 2012.

[14] Changxiao Cai, Sujay Sanghavi, and Haris Vikalo. Structured low-rank matrix factorization for haplotype assembly. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):647–657, 2016.

[15] Serena A Carroll, Jonathan S Towner, Tara K Sealy, Laura K McMullan, Marina L Khristova, Felicity J Burt, Robert Swanepoel, Pierre E Rollin, and Stuart T Nichol. Molecular evolution of viruses of the family filoviridae based on 97 whole-genome sequences. *Journal of virology*, 87(5):2608–2616, 2013.

[16] Mark J Chaisson, Sudipto Mukherjee, Sreeram Kannan, and Evan E Eichler. Resolving multicopy duplications de novo using polyploid phasing. In *International Conference on Research in Computational Molecular Biology*, pages 117–133. Springer, 2017.

[17] Z Chen, B Fu, R Schweller, and B Yang. Linear time probabilistic algorithms for the singular haplotype reconstruction problem from snp fragments. *Journal of Computational Biology*, 15:535–546, 2008.

[18] Zhi-Zhong Chen, Fei Deng, and Lusheng Wang. Exact algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics*, 29(16):1938–45, Aug 2013.

[19] R Cilibrasi, L Van Iersel, S Kelk, and J Tromp. On the complexity of several haplotyping problems. *Algorithms in Bioinformatics*, pages 128–139, 2005.

[20] 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–73, Oct 2010.

[21] The International HapMap Consortium. *The international HapMap project*, 2003.

[22] Fei Deng, Wenjuan Cui, and Lusheng Wang. A highly accurate heuristic algorithm for the haplotype assembly problem. *BMC Genomics*, 14 Suppl 2:S2, 2013.

[23] Francesca Di Giallonardo, Armin Töpfer, Melanie Rey, Sandhya Prabhakaran, Yannick Duport, Christine Leemann, Stefan Schmutz, Nottania K Campbell, Beda Joos, Maria Rita Lecca, et al. Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. *Nucleic acids research*, 42(14):e115–e115, 2014.

[24] EA Duarte, IS Novella, SC Weaver, E Domingo, S Wain-Hobson, DK Clarke, A Moya, SF Elena, JC De La Torre, and JJ Holland. Rna virus quasispecies: significance for viral disease and epidemiology. *Infectious agents and disease*, 3(4):201–214, 1994.

[25] Dawn M Dudley, Matthew T Aliota, Emma L Mohr, Andrea M Weiler, Gabrielle Lehrer-Brey, Kim L Weisgrau, Mariel S Mohns, Meghan E Breit-

bach, Mustafa N Rasheed, Christina M Newman, et al. A rhesus macaque model of asian-lineage zika virus infection. *Nature communications*, 7, 2016.

[26] J Duitama, T Huebsch, G McEwen, Eun-Kyung Suk, and Margret R. Hoehe. Refhap: a reliable and fast algorithm for single individual haplotyping. *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 160–169, 2010.

[27] Jorge Duitama, Gayle K McEwen, Thomas Huebsch, Stefanie Palczewski, Sabrina Schulz, Kevin Verstrepen, Eun-Kyung Suk, and Margret R Hoehe. Fosmid-based whole genome haplotyping of a hapmap trio child: evaluation of single individual haplotyping techniques. *Nucleic Acids Res*, 40(5):2041–53, Mar 2012.

[28] Nicholas Eriksson, Lior Pachter, Yumi Mitsuya, Soo-Yon Rhee, Chunlin Wang, Baback Gharizadeh, Mostafa Ronaghi, Robert W Shafer, and Niko Beerenwinkel. Viral population estimation using pyrosequencing. *PLoS Comput Biol*, 4(5):e1000074, 2008.

[29] P. Fearnhead. Sequential monte carlo methods in filter theory. *Ph.D. dissertation, University of Oxford, Oxford, U.K.*, 1998.

[30] Filippo Geraci. A comparison of several algorithms for the single individual snp haplotyping reconstruction problem. *Bioinformatics*, 26(18):2217–25, Sep 2010.

[31] Abolfazl Hashemi, Banghua Zhu, and Haris Vikalo. Sparse tensor decomposition for haplotype assembly of diploids and polyploids. *bioRxiv*, page 130930, 2017.

[32] Dan He, Arthur Choi, Knot Pipatsrisawat, Adnan Darwiche, and Eleazar Eskin. Optimal algorithms for haplotype assembly from whole-genome sequence data. *Bioinformatics*, 26(12):i183–90, Jun 2010.

[33] Margret R. Hoehe, Karla Kopke, Birgit Wendel, Klaus Rohde, Christina Flachmeier, Kenneth K. Kidd, Wade H. Berrettini, and George M. Church. Sequence variability and candidate gene analysis in complex disease: association of aμ opioid receptor gene variation with substance dependence. *Human Molecular Genetics*, 9(19):2895–2908, 2000.

[34] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[35] Duleepa Jayasundara, Isaam Saeed, Suhinthan Maheswararajah, BC Chang, S-L Tang, and Saman K Halgamuge. Viquas: an improved reconstruction pipeline for viral quasispecies spectra generated by next-generation sequencing. *Bioinformatics*, page btu754, 2014.

[36] Sung Young Jung and Taek-Soo Kim. An agglomerative hierarchical clustering using partial maximum array and incremental similarity com-

putation method. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 265–272. IEEE, 2001.

[37] Govinda M Kamath, Eren Şaşoğlu, and David Tse. Optimal haplotype assembly from high-throughput mate-pair reads. In *2015 IEEE International Symposium on Information Theory (ISIT)*, pages 914–918. IEEE, 2015.

[38] G Lancia, V Bafna, S Istrail, R Lippert, and R Schwartz. Snps problems, complexity, and algorithms. *Proceeding in European Symposium on Algorithms*, pages 182–193., 2001.

[39] Giuseppe Lancia, Vineet Bafna, Sorin Istrail, Ross Lippert, and Russell Schwartz. Snps problems, complexity, and algorithms. In *Algorithms—ESA 2001*, pages 182–193. Springer, 2001.

[40] Eric S Lander and Michael S Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2(3):231–239, 1988.

[41] Adam S Lauring and Raul Andino. Quasispecies theory and the behavior of rna viruses. *PLoS Pathogens*, 6(7), 2010.

[42] Thuy Le, Jennifer Chiarella, Birgitte B Simen, Bozena Hanczaruk, Michael Egholm, Marie L Landry, Kevin Dieckhaus, Marc I Rosen, and Michael J Kozal. Low-abundance hiv drug-resistant viral variants in treatment-

experienced persons correlate with historical antiretroviral use. *PloS one*, 4(6):e6079, 2009.

[43] S Levy, G Sutton, PC Ng, L Feuk, and AL Halpern. The diploid genome sequence of an individual human. *PLoS biology*, 2007.

[44] Heng Li and Richard Durbin. Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[45] Lei M. Li, Jong Hyun Kim, and Michael S. Waterman. Haplotype reconstruction from snp alignment. *Journal of Computational Biology*, 11:505–516, 2004.

[46] Kuo-ching Liang and Xiaodong Wang. A deterministic sequential monte carlo method for haplotype inference. *Selected Topics in Signal Processing, IEEE Journal of*, 2(3):322–331, 2008.

[47] Kuo-Ching Liang, Xiaodong Wang, and Dimitris Anastassiou. A profile-based deterministic sequential monte carlo algorithm for motif discovery. *Bioinformatics*, 24(1):46–55, 2008.

[48] R Lippert, R Schwartz, and G Lancia. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefing in Biioinformatics*, 3:23–31, 2002.

[49] Ross Lippert, Russell Schwartz, Giuseppe Lancia, and Sorin Istrail. Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem. *Briefings in bioinformatics*, 3(1):23–31, 2002.

[50] Serghei Mangul, Nicholas C Wu, Nicholas Mancuso, Alex Zelikovsky, Ren Sun, and Eleazar Eskin. Accurate viral population assembly from ultra-deep sequencing data. *Bioinformatics*, 30(12):i329–i337, 2014.

[51] Miguel Angel Martínez, Gloria Martrus, Elena Capel, Mariona Parera, Sandra Franco, and Maria Nevot. Quasispecies dynamics of rna viruses. In *Viruses: Essential Agents of Life*, pages 21–42. Springer, 2012.

[52] Hirotaka Matsumoto and Hisanori Kiryu. Mixsih: a mixture model for single individual haplotyping. *BMC Genomics*, 14 Suppl 2:S5, 2013.

[53] Abolfazl S Motahari, Guy Bresler, and NC David. Information theory of dna shotgun sequencing. *IEEE Transactions on Information Theory*, 59(10):6273–6289, 2013.

[54] Rasmus Nielsen, Joshua S Paul, Anders Albrechtsen, and Yun S Song. Genotype and snp calling from next-generation sequencing data. *Nat Rev Genet*, 12(6):443–51, Jun 2011.

[55] Wan-Ting Poh, Eryu Xia, Kwanrutai Chin-inmanu, Lai-Ping Wong, Anthony Youzhi Cheng, Prida Malasit, Prapat Suriyaphol, Yik-Ying Teo, and Rick Twee-Hee Ong. Viral quasispecies inference from 454 pyrosequencing. *BMC bioinformatics*, 14(1):1, 2013.

[56] Susana Posada-Cespedes, David Seifert, and Niko Beerenwinkel. Recent advances in inferring viral diversity from high-throughput sequencing data. *Virus Research*, 2016.

[57] Sandhya Prabhakaran, Melanie Rey, Osvaldo Zagordi, Niko Beerenwinkel, and Volker Roth. Hiv haplotype inference using a propagating dirichlet process mixture model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 11(1):182–191, 2014.

[58] Mattia CF Prosperi, Luciano Prosperi, Alessandro Bruselles, Isabella Abbate, Gabriella Rozera, Donatella Vincenti, Maria Carmela Solmone, Maria Rosaria Capobianchi, and Giovanni Ulivi. Combinatorial analysis and algorithms for quasispecies reconstruction using next-generation sequencing. *BMC bioinformatics*, 12(1):1, 2011.

[59] Mattia CF Prosperi and Marco Salemi. Qure: software for viral quasispecies reconstruction from next-generation sequencing data. *Bioinformatics*, 28(1):132–133, 2012.

[60] Sergio Pulido-Tamayo, Aminael Sánchez-Rodríguez, Toon Swings, Bram Van den Bergh, Akanksha Dubey, Hans Steenackers, Jan Michiels, Jan Fostier, and Kathleen Marchal. Frequency-based haplotype reconstruction from deep sequencing data of bacterial populations. *Nucleic acids research*, 43(16):e105–e105, 2015.

[61] E. Punskaya. Sequential monte carlo methods for digital communication. *Ph.D. dissertation, University of Cambridge, Cambridge, U.K.*, 2003.

[62] Michael A Quail, Miriam Smith, Paul Coupland, Thomas D Otto, Simon R Harris, Thomas R Connor, Anna Bertoni, Harold P Swerdlow, and

Yong Gu. A tale of three next generation sequencing platforms: comparison of ion torrent, pacific biosciences and illumina miseq sequencers. *BMC genomics*, 13(1):1, 2012.

[63] Michael G Ross, Carsten Russ, Maura Costello, Andrew Hollinger, Niall J Lennon, Ryan Hegarty, Chad Nusbaum, and David B Jaffe. Characterizing and measuring bias in sequence data. *Genome Biol*, 14(5):R51, 2013.

[64] K Sasirekha and P Baby. Agglomerative hierarchical clustering algorithm–a review. *International Journal of Scientific and Research Publications*, 3(3), 2013.

[65] Melanie Schirmer, Rosalinda D'Amore, Umer Z Ijaz, Neil Hall, and Christopher Quince. Illumina error profiles: resolving fine-scale variation in metagenomic sequencing data. *BMC bioinformatics*, 17(1):125, 2016.

[66] Melanie Schirmer, William T Sloan, and Christopher Quince. Benchmarking of viral haplotype reconstruction programmes: an overview of the capacities and limitations of currently available programmes. *Briefings in bioinformatics*, page bbs081, 2012.

[67] Russell Schwartz. Theory and algorithms for the haplotype assembly problem. *Communications in Information and Systems*, 10:23–38, 2010.

[68] Hongbo Si, Haris Vikalo, and Sriram Vishwanath. Haplotype assembly: An information theoretic view. In *Information Theory Workshop (ITW),*

2014 IEEE, pages 182–186. IEEE, 2014.

[69] Birgitte B Simen, Jan Fredrik Simons, Katherine Huppler Hullsiek, Richard M Novak, Rodger D MacArthur, John D Baxter, Chunli Huang, Christine Lubeski, Gregory S Turenchalk, Michael S Braverman, et al. Low-abundance drug-resistant viral variants in chronically hiv-infected, antiretroviral treatment–naive patients significantly impact treatment outcomes. *Journal of Infectious Diseases*, 199(5):693–701, 2009.

[70] Pavel Skums, Nicholas Mancuso, Alexander Artyomenko, Bassam Tork, Ion Mandoiu, Yury Khudyakov, and Alex Zelikovsky. Reconstruction of viral population structure from next-generation sequencing data using multicommodity flows. *BMC bioinformatics*, 14(Suppl 9):S2, 2013.

[71] Armin Töpfer, Tobias Marschall, Rowena A Bull, Fabio Luciani, Alexander Schönhuth, and Niko Beerenwinkel. Viral quasispecies assembly via maximal clique enumeration. *PLoS Comput Biol*, 10(3):e1003515, 2014.

[72] Armin Töpfer, Osvaldo Zagordi, Sandhya Prabhakaran, Volker Roth, Eran Halperin, and Niko Beerenwinkel. Probabilistic inference of viral quasispecies subject to recombination. *Journal of Computational Biology*, 20(2):113–123, 2013.

[73] Rui Sheng Wang, Ling Yun Wu, Zhen Ping Li, and Xiang Sun Zhang. Haplotype reconstruction from snp fragments by minimum error correction. *Bioinformatics*, 21:2456–2462, 2005.

[74] Ying Wang, Enmin Feng, and Ruisheng Wang. A clustering algorithm based on two distance functions for mec model. *Computational Biology and Chemistry*, 31:148–150, 2007.

[75] Jong Hyun Kim Michael S. Waterman and Lei M. Li. Diploid genome reconstruction of ciona intestinalis and comparative analysis with ciona savignyi. *Genome Research*, 24:1101–1110, 2007.

[76] Kelly Westbrooks, Irina Astrovskaya, David Campo, Yury Khudyakov, Piotr Berman, and Alex Zelikovsky. Hcv quasispecies assembly using network flows. In *Bioinformatics Research and Applications*, pages 159–170. Springer, 2008.

[77] Minzhu Xie, Jianxin Wang, and Tao Jiang. A fast and accurate algorithm for single individual haplotyping. *BMC Syst Biol*, 6 Suppl 2:S8, 2012.

[78] Osvaldo Zagordi, Arnab Bhattacharya, Nicholas Eriksson, and Niko Beerenwinkel. Shorah: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC bioinformatics*, 12(1):119, 2011.

[79] Osvaldo Zagordi, Lukas Geyrhofer, Volker Roth, and Niko Beerenwinkel. Deep sequencing of a genetically heterogeneous sample: local haplotype reconstruction and read error correction. *Journal of computational biology*, 17(3):417–428, 2010.

[80] Osvaldo Zagordi, Rolf Klein, Martin Däumer, and Niko Beerenwinkel.

Error correction of next-generation sequencing data and reliable estimation of hiv quasispecies. *Nucleic acids research*, 38(21):7400–7409, 2010.

[81] Yu Ying Zhao, Ling Yun Wu, Ji Hong Zhang, Rui Sheng Wang, and Xiang Sun Zhang. Haplotype assembly from aligned weighted snp fragments. *Computational Biology and Chemistry*, pages 281–287, 2005.

Vita

Soyeon Ahn received the B.S. degree in Electrical Engineering from Korea University, South Korea, in 2008 and the M.Eng. degree in Electrical Engineering from Korea Advance Institute of Science and Technology(KAIST), South Korea, in 2010. She worked as a system engineer at Samsung Thales, South Korea, between 2010 and 2011. She is currently pursuing the Ph.D. degree in Electrical and Computer Engineering from the University of Texas at Austin. She was the recipient of the honors scholarship at Korea University, (2003-2006), science and engineering scholarship(2004), national science fellowship (2007), the national fellowship for graduate students at KAIST (2008-2009), and the MCD fellowship from the Cockrell School of Engineering at the University of Texas at Austin (2012-2013). Her research interests include bioinformatics, machine learning, and statistical signal processing.

Permanent address: 9905 Chukar Circle
Austin, Texas 78758

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.