INFERENCE OF VIRAL STRAINS USING METAGENOMICS DATA

By

Jiao Chen

A DISSERTATION

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Computer Science—Doctor of Philosophy

2018

ABSTRACT

INFERENCE OF VIRAL STRAINS USING METAGENOMICS DATA

By

Jiao Chen

RNA Viruses, such as human immunodeficiency virus (HIV), influenza, and hepatitis C virus (HCV), have great impact on human health. The high mutation rate of RNA viruses can produce a population of different but closely related virus sequences called viral quasispecies. To develop prevention and treatment strategies for viral pathogens, an essential and fundamental step is to characterize their sequences and abundances at the strain-level for a viral quasispecies.

Advances in next-generation sequencing (NGS) technologies have opened up new opportunities to study viruses. Viral metagenomic data, which contains the genetic information for a bunch of viruses in the same habitat, have become the major resource for characterizing RNA viruses. Although there are many pipelines for analyzing viruses in metagenomic data, they usually lack three functions. First, novel viruses or viruses that lack closely related reference genomes cannot be detected with high sensitivity. Second, strain-level analysis is usually missing. Although there are several assembly tools specifically designed for viral haplotypes, *de novo* assembly of virus genomes is still a computationally challenge task due to the error-prone short reads, high similarities between related strains, and unknown number or abundance of virus haplotypes. Third, it is hard to estimate the number of haplotypes and their abundances in a quasispecies.

In this dissertation, we have developed a pipeline with three tools, TAR-VIR PEHaplo and VirBin to address the challenges existing for viral metagenomic assembly and analysis. TAR-VIR is a tool for classifying and enriching viral reads from metagenomic data without relying on complete or high-quality reference genomes. It is optimized for identifying RNA viruses

from metagenomic data with an efficient Burrows-Wheeler Transform (BWT) based overlapping method. TAR-VIR was tested on both simulated and real viral metagenomic datasets. The results demonstrated TAR-VIR competes favorably with benchmarked tools. PEHaplo is a *de novo* haplotype reconstruction tool, which employs paired-end reads to distinguish highly similar strains for viral quasispecies data. It was applied on both simulated and real quasispecies data, and the results were benchmarked against several recently published *de novo* haplotype reconstruction tools. The comparison shows that PEHaplo outperforms the benchmarked tools in a comprehensive set of metrics. With assembled viral contigs, VirBin focuses on estimating the number of haplotypes and clustering the contigs to different haplotypes. VirBin firstly identifies windows from contigs alignment profile to estimate haplotype number and better calculate the contig abundances. Then, it applies an Expectation-Maximization method to cluster the contigs based on contig abundance levels. The experimental results of VirBin on both simulated and real data sets show that the window-based method can precisely estimate the haplotype number, and generate more accurate abundance estimation for contigs that will eventually lead to superior clustering results. In addition, this dissertation also contains one chapter for the other work we have done for identifying the primary transcription start sites for miRNA genes in Caenorhabditis elegans and mouse with Cap-seq data.

ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Dr. Yanni Sun. Her perfect supervision has been guiding me through all my research and dissertation work during my Ph.D. study. Nothing could be achieved without her devoted time, patience, and inspiration. During the past five and half years studying at Michigan State University, Dr. Sun helped me on my research work, courses, and how to write or present research results. I have been kept improving in making presentations to audiences from different backgrounds, modeling bioinformatic problems with computational techniques, and reading or writing papers.

Next, I want to thank my other committee members Dr. David Arnosti, Dr. Kevin Liu, and Dr. Jin Chen. Dr. Arnosti carefully read and provided me with many valuable suggestions for my first paper on miRNAs. He also invited me to join and talk on Journal Club, which was an important part of my Ph.D. training. Dr. Liu gave me the lecture of CSE836 (Computational Biology) on my second-year study in MSU, which is a course closely related to my Ph.D. research. He also provided us with several chances of presenting papers. Dr. Chen helped me a lot for either my comprehensive exam or final defense. I also thank Dr. Jianrong Wang. He had been helping me on the binning project by providing helpful discussions and being always approachable.

I also thank my lab mates Dr. Yuan Zhang, Dr. Cheng Yuan, Dr. Jikai Lei, Dr. Rujira Achawanantakun, Nan Du and Yumeng Wen, for their help on my either research work or daily life. My thank also goes to my previous roommate Chaoyue Liu and Shaohua Yang.

Finally, I want to express my sincere gratitude to my parents and sister. They have always been encouraging and supporting me for pursuing my career, and are always there whenever I need them.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ALGORITHMS

Chapter 1

Introduction

Many of most important human viral pathogens, such as human immunodeficiency virus (HIV), hepatitis C virus (HCV), Severe Acute Respiratory Syndrome (SARS) coronavirus (SARS-CoV), and H1N1 flu virus, are highly variable RNA viruses. They usually mutate quickly in infected cells or organisms and exist as a collection of different but closely related viral genomes with dynamic distribution, which is referred to as viral quasispecies [41, 3]. The quasispecies dynamics provides RNA viruses with high adaptive potential to overcome internal or external selective constraints, such as immune responses, antiviral agents, etc., making disease prevention and treatment difficult. These viruses still claim the lives of millions each year despite centuries studies of the vaccine and treatment [158, 140]. Thus, characterizing human viral pathogens, including recognizing novel ones, remains crucial.

Development of next-generation sequencing (NGS) technologies sheds light on characterizing the viral genomes and their abundances in environment. The deep sequencing data of microbial communities (metagenomic data) contains the genetic information of the microbes living in the same habitat and has become the primary source for virus discovery [165]. Viral metagenomic data has significant advantages over traditional methods since it is unnecessary to isolate viral species and cultivate them in the laboratory. Also, multiple pathogens, including new ones, can be identified in a single assay. Currently, the most popular NGS strategies for metagenomics, such as Illumina MiSeq and HiSeq, are cost-effective and with high-throughput. They can produce billions of short, highly accurate 100-300 bp reads within a few days for large-scale microbiome research

projects [109].

While with these advantages, assembling microbial genomes and characterizing their compositions from metagenomic data meet with several computational challenges. The first challenge is the sheer data size from high-throughput sequencing, which requires massive computational resources for assembling and analyzing the microbes. The second one is the short read length, which makes it difficult to resolve the repeated sequences within the same organism, or shared between different organisms. Third, it is difficult to assemble the low abundance microbial genomes with the heterogeneous coverages for distinct microbes. In addition to these well-known challenges, reconstructing RNA viruses in metagenomics faces several unique obstacles. First, the references of viruses are still very limited. Usually, only a small percentage of reads in metagenomic data can be mapped to available viral genomes, either because the viruses are mutating quickly or are novel ones. Characterizing RNA viruses without quality reference genomes is thus needed. Second, most clinically important viruses are often fast mutating RNA viruses that can lead to viral populations of high genetic diversity (quasispecies) [107]. As different genomes in the same viral population can have different properties such as resistance to anti-viral drugs, distinguishing these different but highly similar viral strains (haplotypes) is essential but computationally difficult. Third, there are usually only small amounts of eukaryotic viruses in viral metagenomic data. Although filtration techniques and amplification methods may be applied to enrich RNA viruses, these strategies may cause various types of biases. Thus, effective preprocessing methods are still important to identify reads for the targeted viruses.

In this work, to tackle all these challenges for characterizing viral haplotypes and their abundances from metagenomic data, we have proposed a pipeline with three tools to (1) enrich target viruses' reads with low-quality or partial reference sequences; (2) *de novo* assembly of the viral haplotypes from the enriched reads; (3) analyze haplotype abundances by binning assembled

contigs to different groups.

In the following part of this chapter, I will first give a brief introduction to the pipeline I have developed during my Ph.D. study, and then introduce the necessary background for next-generation sequencing and viral quasispecies theory.

1.1 Overview of the pipeline

As shown in Figure 1.1, there are three main components for this pipeline. In the first step, we utilize an overlap-based extension method to isolate and enrich the reads for target viruses before assembly. With the big scale of viral metagenomic data, there are usually a small proportion of viral reads among them, with many contaminations from host genomes or phages. Directly assembling viruses from the metagenomic data is difficult and time-consuming. Therefore, it is important to isolate and enrich the viral reads we are interested in before assembly. This part is detailed in Chapter 2. Second, we assemble the different viral haplotypes from the enriched viral reads with a paired-end information incorporated overlap graph. The details for this part are in Chapter 3. Third, with the assembled viral contigs, we use an Expectation-Maximization algorithm to estimate the abundances of different viral haplotypes by clustering the assembled contigs to different groups, where each group represents one haplotype. The details for this part are in Chapter 4. Finally, in Chapter 5, I introduce the other work I have done for studying the transcriptional regulations of miRNA genes with Cap-seq data.

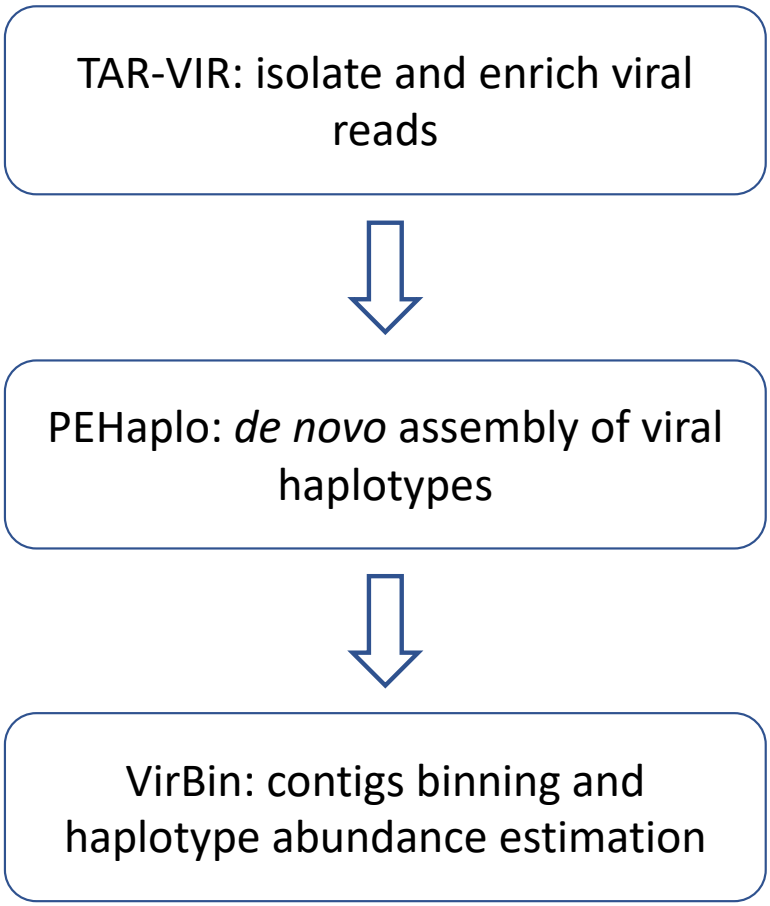

Figure 1.1: Pipeline overview. (1) TAR-VIR for enriching viral reads of interest from metagenomic data. (2) PEHaplo for *de novo* assembly of viral haplotypes. (3) VirBin for binning assembled contigs and estimate the haplotype abundances.

1.2 Next-generation sequencing

Though the automated Sanger sequencing technology has proved its power with a number of monumental accomplishments, including the completion of the Human Genome Project (HGP), its limitations showed a need for new and improved technologies for large-scale and low-cost sequencing [98]. As the automated Sanger method is considered to be the 'first-generation' technology, newer methods are referred to as next-generation sequencing (NGS). While NGS technologies constitute various strategies and are quite diverse in sequencing biochemistry, their workflows are conceptually similar. These methods have similar basic mechanisms and are classified as cyclic-array sequencing, which can be summarized as the sequencing of a dense array of DNA features by iterative cycles of enzymatic manipulation and imaging-based data collection[101]. The basic protocol of NGS technologies includes template preparation, sequencing and imaging, and data analysis. The major advance of NGS is the ability to produce large volume of data with low cost - in some cases over one billion short reads per instrument run[98].

There are several commercially available technologies, including Roche/454, Illumina/Solexa, Life/APG and Helicos BioSciences, etc. The unique combination of specific protocols distinguishes one technology from another and determines the type of data produced from each platform. Here we will give a detailed introduction to Illumina/Solexa sequencing technology and analyze the different computational techniques for dealing with the data.

The Illumina/Solexa sequencing technology is called sequencing by synthesis (SBS). There are four basic steps for Illumina NGS workflows:

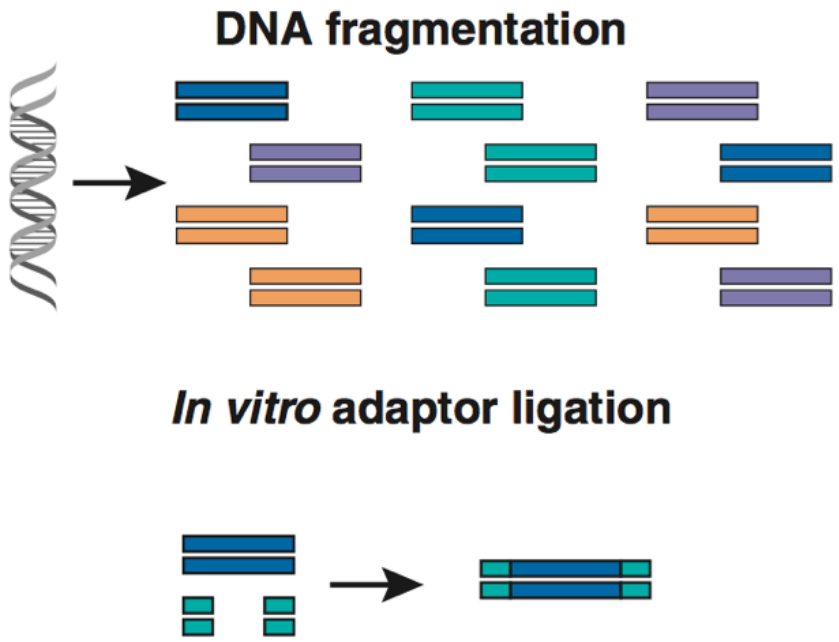

Figure 1.2: Library preparation. Genomic DNA is fragmented and common adaptors are ligated to both ends of them.

1.2.1 Library preparation

Before sequencing the DNA or cDNA sample, the long DNA sequences are first randomly broken into smaller sizes by sonication. These short DNA fragments produced cannot be sequenced directly; adaptors are ligated to the 5' and 3' ends of these DNA fragments from which either fragment templates or mate-pair templates are created (Figure 1.2). Adapters are short oligonucleotides which are attached to the DNA to be sequenced. An adapter can provide a priming site for both amplification and sequencing. All the DNA templates are called a sequencing library.

1.2.2 Cluster Generation

While bases of a DNA sequence are detected by fluorescence, most imaging systems have not been designed to detect single fluorescent events. Amplification of templates are required. Illumina applies a method called solid-phase amplification [43]. High-density forward and reverse primers

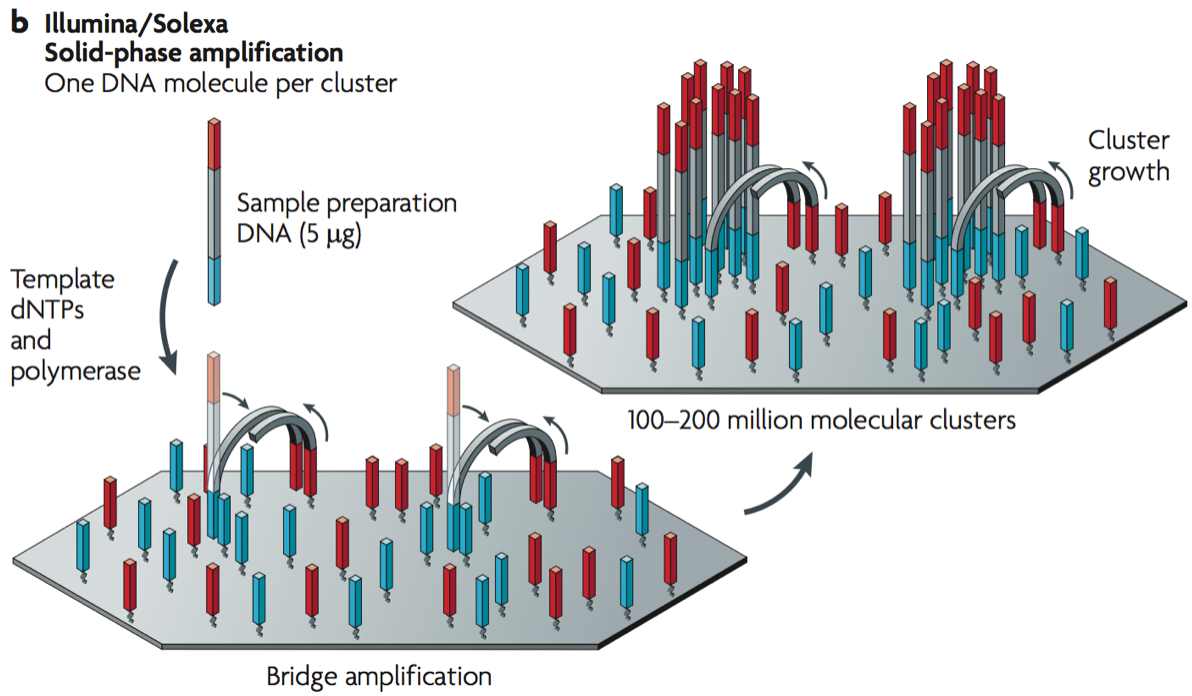

Figure 1.3: Solid-phase amplication. Each single-stranded, single-molecule template is captured by primer and immobolize to the solid surface. Each bound template is amplified into a clonal cluster through bridge amplication.

are covalently attached to a glass slide, and the ratio of the primers to the template defines the surface density of the amplified clusters (Figure 1.3). The library is loaded into a flow cell where fragments are captured by the primers. Each fragment is then amplified into distinct, clonal clusters through bridge amplification. When cluster generation is complete, the free ends of templates can be hybridized by a universal sequencing primer to initiate the NGS reaction.

1.2.3 Cyclic Reversible Termination (CRT) Sequencing

Clonal amplification results in a population of identical templates, each of which has undergone the sequencing reaction. Upon imaging, the observed signal is a consensus of the nucleotides added to the identical templates for a given cycle.

Illumina technology utilizes a proprietary reversible terminator-based method that detects sin-

gle bases as they are incorporated into DNA template strands. In the first step, a DNA polymerase binds to the primed template and adds or incorporates just one fluorescently modified nucleotide, which represents the complement of the template base. A reversible terminator is on every modified nucleotide to prevent multiple additions in one round. Following incorporation, the remaining unincorporated nucleotides are washed away. Using the four-colour chemistry, each of the four bases has a unique emission. After each round, imaging is used to determine the identity of the incorporated nucleotide. A cleavage step is followed to remove the terminating/inhibiting group and the fluorescent dye. The next incorporation step will start after an additional washing (Figure 1.4).

1.2.4 Data analysis

After NGS reads have been generated, they are aligned to a known reference sequence or assembled de novo [149, 72, 70, 22, 83, 123]. The production of large numbers of low-cost reads makes the NGS platforms useful for many applications. These include variant discovery by resequencing targeted regions or whole genomes, such as detecting single nucleotide polymorphisms (SNPs) or structural variants (SVs); cataloguing the transcriptomes of cells, tissues and organisms (RNA-seq) [155]; genome-wide profiling of epigenetic marks and chromatin structure (ChIP-seq, MNase-seq, etc.) [138, 113]; and species classification and/or gene discovery by metagenomics studies [120].

1.2.5 Paired-End Sequencing

Paired-end sequencing sequences both ends of long DNA fragments in a sequencing library and aligning the forward and reverse reads as read pairs (Figure 1.5). In addition to producing twice the number of reads for the same time and effort in library preparation, read pairs from two ends of DNA fragments enable more accurate read alignment and the ability to overcome repetitive

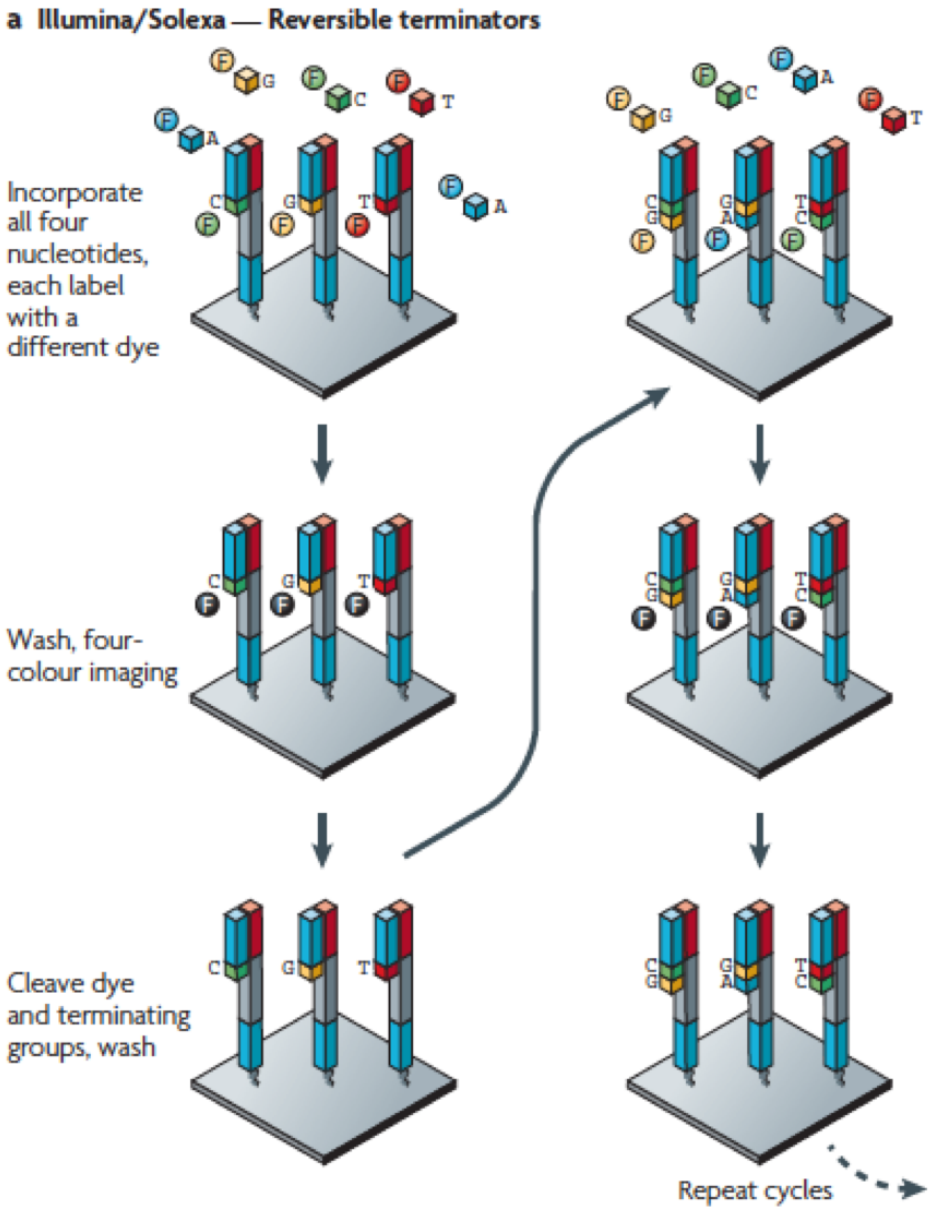

Figure 1.4: The Illumina/Solexa four-colour cyclic reversible termination (CRT) method. It uses 3'-O-azidomethyl reversible terminator chemistry on solid-phase-amplified template clusters. After imaging, a cleavage step removes the fluorescent dyes and regenerates the 3'-OH group for next-round incorporation.

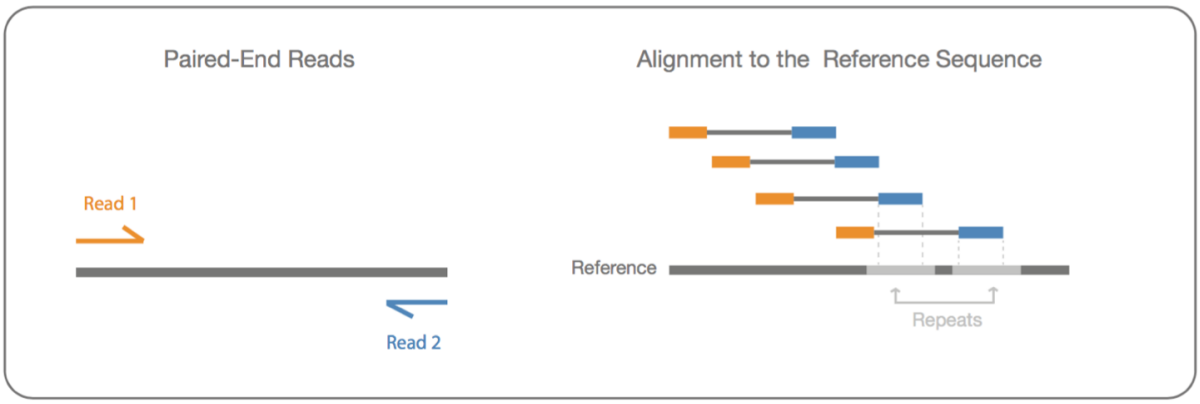

Figure 1.5: Paired-end sequencing enables both ends of the DNA fragment to be sequenced. The distance between each paired reads is known and alignment algorithms can take advantage of to map the reads over repetitive regions more precisely.

regions.

Aligning reads within repetitive regions or in corresponding regions that may not exist in the reference genome is difficult. The repeats and gaps in the reference genome can also cause abnormal extremely high peaks in the mapping profile, which may lead to false positives for downstream functional analysis [25]. Paired-end reads can resolve the correct genome assignment for some repetitive regions as long as one read in the pair is unique to the genome [98].

1.3 Virus quasispecies

A vast number of medically important viruses, including human immunodeficiency virus (HIV), hepatitis C virus (HCV), and influenza, are RNA viruses. They replicate with extremely high mutation rates and exhibit significant genetic diversity ,which allow them to rapidly adapt to dynamic environments and evolve resistance to vaccines or antiviral drugs [75]. The RNA virus evolution and dynamics can be modeled by quasispecies theory. A quasispecies is a cloud of diverse variants that are genetically linked through mutations, interact cooperatively on a functional level and

collectively contribute to the characteristics of the population.

1.3.1 The quasispecies theory

Imagine a quasispecies with an infinitely large population of different strains, each carrying a genome of length L. Let x_i denotes the relative abundance/frequency of strains i. We have $\sum_{i=0}^{n} x_i = 1$. The genomic structure of the population can be characterized by the vector $\vec{x} = (x_0, x_1, ..., x_n)$.

Denote by f_i the fitness of genome i, which means genomes of type i are being reproduced at rate f_i. The fitness landscape is given by the vector $\vec{f} = (f_0, f_1, ..., f_n)$. The average fitness of the population is the inner product of the vectors $\vec{x}$ and $\vec{f}$, $\phi = \vec{x}\vec{f}$.

Mistakes can happen when a genome replicates. We use mutation matrix $\mathbf{Q} = [q_{i,j}]$ to denote the probability that replication of genome i results in genome j. $\mathbf{Q}$ is a stochastic matrix, with which each row is a probability and each row sums to one, $\sum_{j=0}^{n} q_{ij} = 1$.

The quasispecies equation is given by [107]

$$x_i' = \sum_{j=0}^{n} x_j f_j q_{ji} - \phi x_i \tag{1.1}$$

where x_i' is the differentiation of x_i with respect to time. Sequence i is obtained by replicating any sequence j at fitness rate f_j times the probability that replication of sequence j generates sequence i. To ensure that the total population size remains constant, that is $\sum_{i=0}^{n} x_i' = 0$, each sequence is removed at rate ϕ (Proof: $\sum_{i=0}^{n} x_i' = \sum_{j=0}^{n} x_j f_j q_{ji} - \phi \sum_{i=0}^{n} x_i = \sum_{i=0}^{n} q_{ji} \sum_{j=0}^{n} x_j f_j - \phi = \sum_{j=0}^{n} x_j f_j - \phi = 0$).

If we combine the fitness landscape, $\vec{f}$, and the mutation matrix, $\mathbf{Q}$, we can obtain the mutation-selection matrix,

$$\mathbf{W} = [w_{ji}] = [f_j q_{ji}] \tag{1.2}$$

With the mutation-selection matrix, the quasispecies equation can be written in vector notation as

$$\vec{x}' = \vec{x}\mathbf{W} - \phi\vec{x} \tag{1.3}$$

Therefore the equilibrium of quasispecies dynamics is given by

$$\vec{x}\mathbf{W} = \phi\vec{x} \tag{1.4}$$

This is a standard eigenvalue problem, where the average fitness ϕ is the largest eigenvalue of the matrix $\mathbf{W}$ and $\vec{x}$ is the eigenvector associated with this eigenvalue. Thus, the eigenvector $\vec{x}^*$ of mutation-selection matrix $\mathbf{W}$ provides the equilibrium structure of the quasispecies, with the proper normalization $\sum_{i=0}^{n} x_i = 1$.

Error catastrophe

The quasispecies equation 1.1 describes the movement of a population through sequence space. The quasispecies attempts to climb uphill in the mountain range of the fitness landscape and reach local or global peaks (Figure 1.6). The error threshold plays a key role in the success of the evolutionary walk.

When the mutation rate μ is above the error threshold, the ability of the quasispecies to climb uphill and to remain on top of a mountain peak is impaired [42, 13]. Small increases in mutation rate will upset this balance as the master sequence itself disappears and meaningful genetic information is lost in an avalanche of errors. Early studies with vesicular stomatitis virus (VSV) showed that chemical mutagens generally reduced viral infectivity, and studies with poliovirus demonstrated that mutagenic nucleoside analogs push viral populations to extinction - a 4-fold increase in mutation rate resulted in a 95% reduction in viral titer [52, 153].

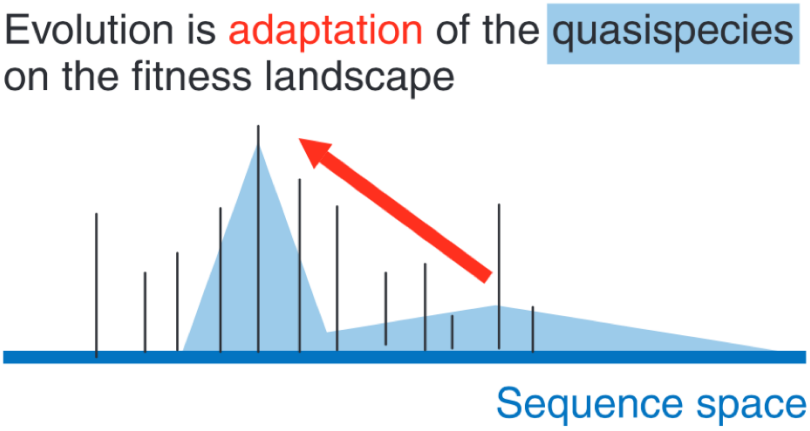

Figure 1.6: Quasispecies adaption. Quasispecies tends to go uphills in high-dimensional fitness landscape. The x-axis is the sequence space and the y-axis is the fitness. The higher they get, the fitter they are.

Chapter 2

TAR-VIR: TARgeted VIRal reads classification and strain reconstruction from metagenomic data

2.1 Background

Pathogenic human viruses such as human immunodeficiency virus (HIV), hepatitis C virus (HCV), Severe Acute Respiratory Syndrome (SARS) coronavirus (SARS-CoV), and H1N1 flu virus, still claim the lives of millions each year despite centuries studies of the vaccine and treatment [158, 140]. Thus, characterizing human viral pathogens, including recognizing novel ones, remains crucial. The deep sequencing data of microbial communities (metagenomic data) contains the genetic information of the microbes living in the same habitat and has become the primary source for virus discovery [165]. For example, Li et al. [80] conducted comprehensive virus screening in plasma samples of 19 antiretroviral-treated HIV patients using metagenomic sequencing. Lim et al. [85] studied the dynamics of eukaryotic RNA and DNA viruses for the first year infants. There are also global-scale studies on viruses in natural environmental samples such as ocean water [102, 126]. Viral metagenomic data has significant advantages over traditional methods since it is unnecessary to isolate viral species and cultivate them in the laboratory. Also, multiple

pathogens, including new ones, can be identified in a single assay.

We are particularly interested in characterizing RNA viruses via metagenomic sequencing because many of them are clinically important and there are still urgent needs for better prevention and treatment strategies. In addition to some well-known challenges for metagenomic assembly such as sheer data size, short read length, and heterogeneous coverage, reconstructing RNA viruses in metagenomics faces several unique obstacles. First, the references of viruses are still very limited. Usually, only a small percentage of reads in metagenomic data can be mapped to available viral genomes, either because the viruses are mutating quickly or are novel ones. Characterizing RNA viruses without quality reference genomes is thus needed. Second, most clinically important viruses are often fast mutating RNA viruses that can lead to viral populations of high genetic diversity (quasispecies) [107]. As different genomes in the same viral population can have different properties such as resistance to anti-viral drugs, distinguishing these different but highly similar viral strains (haplotypes) is important but computationally difficult. Third, there are usually only small amounts of eukaryotic viruses in viral metagenomic data. Although filtration techniques and amplification methods may be applied to enrich RNA viruses, these strategies may cause various types of biases. Thus, effective preprocessing methods are still important to identify reads for the targeted viruses.

In this study, our goal is to develop a new pipeline that can classify RNA viral reads with high sensitivity and also produce the assembled viral strains (i.e. haplotypes) from metagenomic data.

2.1.1 Related work

There exist some pipelines for conducting the composition and functional analysis of viral metagenomic data [99, 128, 105, 122]. Depending on the required inputs, existing programs can be roughly divided into two groups.

Some viral metagenomic analysis tools take metagenomic assemblies as input and conduct reference-based taxonomic and functional analysis. For example, Espino et al. [112] identify viral sequences from assembled metagenomic contigs of sizes greater than 5kb. VirSorter [127] detects viruses in assembled contigs at least 3kb. The viral sequences are usually screened by comparing the contigs with a curated set of viral protein families.

Although virus detection using assembled contigs is usually more sensitive than using short reads, conducting *de novo* assembly for raw viral metagenomic data requires massive computing resources. Also, applying generic *de novo* assembly tools directly to metagenomic data can lead to short or chimeric contigs due to the heterogeneous coverage along the genomes in metagenomic data and high similarity between strains. Therefore, most of the assembly methods for viral metagenomics combine reference-based classification and *de novo* assembly. This strategy usually classifies reads into different taxonomies or functional groups using reference-based methods and then conduct *de novo* assembly for reads within the same group. For example, VIP [84], drVM [86], and VirusTAP [162] all apply this strategy. They classify/enrich viral reads by either aligning reads to available viral references or removing host and other unrelated microbial reads. Next, existing assembly tools such as SPAdes [8] are employed to the virus-like reads to produce the final assembly results. While these tools made significant contributions in purifying the data by removing non-virus reads and then classifying virus-like reads into functional/taxonomical groups, their performance heavily depends on the quality of the references.

For RNA virus characterization, related tools also include haplotype reconstruction pipelines designed to assemble viral strains in a quasispecies. A majority of these tools are reference-based and take the alignments of reads against reference genomes as input. HaploClique [147], ViQuas [59], VGA [95] all belong to this group.

In addition, there are generic assembly tools that apply extension strategies to iteratively extend

the "seed" contigs. For example, PRICE [129] takes advantage of paired-end reads to iteratively align reads on the initial contigs and extend them.

The limitation of reference genomes is the critical challenge of applying the above reference-based tools for RNA viruses analysis in metagenomic data. While regarded as the most abundant biological entities on earth, only a small portion of viruses have been sequenced and characterized. Besides, for RNA viruses with high mutation rates, high-quality reference genomes of a viral population are not always available. For example, many emerging viral diseases are caused by zoonotic viruses, which originate in vertebrates but can infect humans. The genomes of some emerging viruses may only share medium sequence identity with their peers in animals, creating difficult circumstances for reference-based virus reads classification.

2.1.2 Overview of our work

Here we introduce TAR-VIR, which provides a useful addition to existing tools for identifying targeted RNA viruses and their haplotypes in metagenomic data. The "targeted" viruses are those that still possess local sequence similarity with their homologs in the reference genomes. A completely new virus that does not share any conservation with any reference genome won't be detected by our method.

Our pipeline combines reference-based strategy and *de novo* assembly and is optimized for the following applications. 1) Identifying host-switching viruses such as SARS-CoV using remotely related viruses in other hosts as the references. 2) Reconstructing viral haplotypes that are divergent from a known virus family. 3) Recovering viruses and their genomes that contain genes or functional sites of interest to users. TAR-VIR is faster and more effective in identifying targeted viruses than generic assembly, which is particularly important for large and complicated metagenomic datasets containing a small percentage of viruses. Meanwhile, TAR-VIR is more tolerant to

incomplete or low-similarity references than existing reference-based tools.

We applied TAR-VIR to a simulated metagenomic data set containing five haplotypes of SARS-CoV and a real human blood plasma metagenomic data set. The benchmarked results with both *de novo* assembly tools and reference-based haplotype reconstruction tools demonstrated the utility of TAR-VIR in recovering RNA viruses from metagenomic data with limited references.

2.2 Methods

Following a stand-alone error-correction step, our pipeline performs the following three steps. *First*, we construct the set of "seed reads" by mapping the reads against provided reference sequences, which could be reference genomes in available databases or functional sites such as genes. All the reads that can be mapped to the reference constitute the set of "seed reads". *Second*, we recruit reads that form significant overlaps with the seed reads. Newly recruited reads will be added to the seed set. This process will iterate until no new reads can be recruited. *Third*, we conduct strain-level assembly using the reads identified in the second step.

2.2.1 Two scenarios

The above pipeline is visualized for two scenarios in Figure 2.1. In scenario 1, users are trying to detect viruses that contain a functional site, such as a gene. Unlike the well-studied gene-centric assembly in metagenomic data, our goal is to recover the whole genome that contains a particular gene. In this method, the gene is provided as a reference, and reads mapped to it are the seed reads. Overlap detection is then applied to recruit more reads that belong to the same viruses as the seed reads. The read recruitment process is presented in Figure 2.1(A).

In scenario 2, the goal is to identify viruses that lack quality reference genomes. This is partic-

ularly important for host-switching viruses, which may not always conserve high sequence similarity with their related peers in other hosts. For example, SARS-CoV shares about 80% of sequence identity with the bat coronavirus according to BLAST [20]. And the identity is lower than 50% at different loci. Thus, conventional read mapping methods cannot capture all reads from the targeted viruses when they lack high similarity with the available references. Figure 2.1(B) presents the process of identifying reads of the target virus with a remotely related virus as the reference. Although the mapped reads are scattered along the reference genome with low coverage, sufficient reads belonging to the target virus can be recruited through overlap detection.

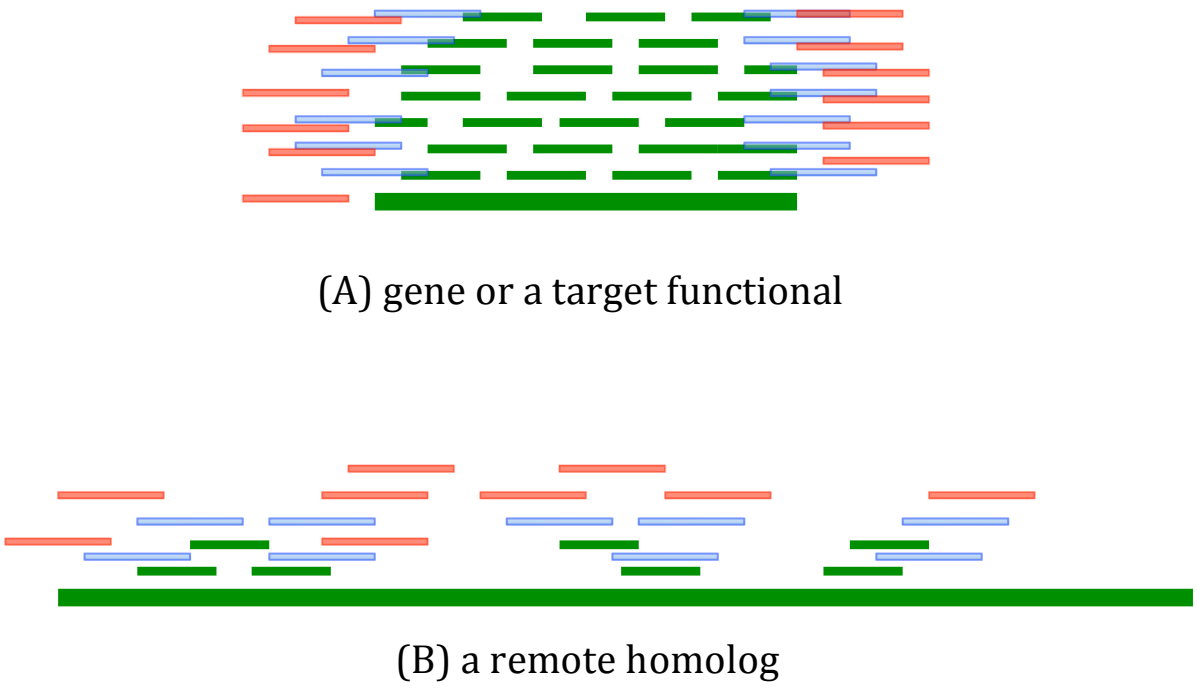

Figure 2.1: Two scenarios. (A). The reference is a gene or other functional site (long green bar). The reads are represented by short lines. Short green lines can be mapped to the reference sequence and define the set of seed reads. The first iteration of overlap detection will identify new reads (blue lines) overlapping with the seed reads. The second iteration of overlap detection will identify more reads (red lines). (B). The reference is a remotely related genome (long green bar). The seed reads can be mapped to the reference genome and are represented by short green lines. Two iterations of overlap detection will recruit new reads (blue lines and red lines, respectively).

2.2.2 Validity of read recruitment using overlap detection

In this section, we will conduct careful analysis to examine whether using overlaps will be both sensitive and accurate for classifying reads in the same quasispecies. In addition, we will use the analysis to determine the appropriate overlap size.

An ideal read recruiting process should only capture the reads from the viruses of interest. At the same time, it should capture all reads belonging to the targeted viral populations (i.e., quasispecies). The success of the overlap extension-based read classification depends on the quality of seed reads, sequencing depth of the targeted viruses, and sequence similarity between different microbes. In particular, if many microbes share long common regions with the targeted viruses, the overlap extension will recruit a large number of reads from unrelated species.

As any read that forms an overlap of size above a given threshold τ with a seed read is recruited, contamination, which refers to reads not sequenced from the targeted viral populations, can be checked by examining whether genomes of different viruses share common regions with sizes above τ. Therefore, we computed the sizes of longest common substrings (LCSs) between different viruses. The LCSs between viruses and other microbial species were also examined. The details for LCS calculation can be found in Supplementary Materials Section 1. The results are shown in Supplementary Figure S1(A-C).

In summary, the sizes of LCSs between different viral genomes or between human viruses and bacteria are usually smaller than 100 bp. LCSs longer than 100 bp are mostly between viruses from the same genus or different genotypes of the same virus. For example, Vaccinia virus and Variola virus share an LCS of 469 bp, and HCV genotype 7 and HCV genotype 5 share an LCS of 154 bp.

Meanwhile, it is also necessary to evaluate whether reads belonging to the same quasispecies

can be recruited using overlap detection. As the characterized haplotypes for different RNA viruses are very limited, instead of computing the LCS using available data, we estimated the LCSs within a quasispecies using a probability model. With the mutation rate μ at each base during virus replication, the probability distribution of LCS length between two viral strains that are n generations apart can be calculated with dynamic programming [26]. As an example, the probability distribution of LCS sizes between two HIV strains is shown in Supplementary Figure S1(D).

The result reveals that the LCSs between different haplotypes of the same viral population are usually much longer than LCSs between different viruses or an Illumina read size. Thus, even with the initial seed reads aligned to only one haplotype, the reads of other haplotypes can be recruited through the long common regions shared by different haplotypes. The reads sequenced from the common regions act like baits to recruit reads from different haplotypes.

The short LCSs between different viruses and the long LCSs between different haplotypes within the same quasispecies enable high sensitivity and specificity of the read recruiting process. By choosing a proper overlap threshold using the derived LCS size distribution, we are able to recruit reads for the same viral quasispecies without introducing contamination from other microbes.

2.2.2.1 Sequencing errors

Without considering the sequencing errors, uneven sequencing coverage, and the virus recombination, the above analysis for viral quasispecies provides an upper bound of reads' overlap sizes. Sequencing errors will shorten overlaps between reads and may prevent recruiting all reads belonging to the same quasispecies. To recruit sufficient reads for assembly, we can construct either approximate overlaps by allowing mismatches/gaps or exact overlaps on error-corrected reads. Considering the risk of introducing contaminations by approximate overlap and extensive research in error correction field, we chose to use stand-alone error correction tools paired with exact over-

lap detection. Low sequencing depth will cause insufficient overlaps and thus affects the performance of read recruitment. There is the possibility the low coverage regions of a genome cannot be assembled.

2.2.2.2 Chimeric reads

One recent study revealed that chimeric reads, which contain sequences from more than one species, can be generated *in vitro* during the preparation of high-throughput sequencing libraries [114]. These chimeras may have overlaps with more than one species, thus introducing contaminations from the host or unrelated microbes. In our experiments, we set the overlap threshold longer than half of the read size to prevent recruiting these chimeric reads or extending from them.

2.2.3 Read recruiting

Let r_i and r_j be two reads. If there is a proper suffix of r_i that is the prefix of r_j or vice versa, r_i and r_j form an overlap. In practice, we will also account for the overlaps formed by r_i and r'_js reverse complement. Naive algorithm for finding overlaps takes quadratic time and thus will not be able to scale to large data sets. There are a few data structures and methods available for efficient overlap detection [47, 142]. We apply the methods with BWT and FM-index [142] for efficient search. In the first step, all reads are concatenated into a single sequence $T[1..n]$ using \$ as a delimiter, where n is the number of reads in T. Then, multi-key "quicksort" is applied to sort all the suffixes of T for constructing a generalized suffix array $SA(T)$ [121]. Then $BWT(T)$ can be constructed using the following equation, where $BWT[i]$ and $SA[i]$ are abbreviated representations of $BWT(T)[i]$ and

$SA(T)[i]$, respectively.

$$BWT[i] = \begin{cases} T[SA[i]-1], & \text{if} \quad SA[i] > 0 \\ \$, & \text{if} \quad SA[i] = 0 \end{cases} \tag{2.1}$$

With T and $BWT(T)$, the backward search can be used to detect overlaps between a query read and all other reads. After matching τ (the overlap threshold) characters, we search for the delimiter '$' to find the prefixes overlapping with the query's suffix.

The backward search algorithm for BWT needs to use two auxiliary data structures built on T and BWT. One data structure is an array $C[1,\ldots,|\Sigma|]$. For each character c in the alphabet Σ, $C[c]$ contains the total number of character in T that are alphabetically smaller than c (including repeating characters). The second data structure is a two-dimensional array $Occ(c,i)$, where $c \in \Sigma$ and $1 \leq i \leq |T|$. For the ith position in BWT, $Occ[c][i]$ is the number of occurrences of c in prefix $BWT[1\ldots i]$. Examples of the SA, BWT, C, and Occ are provided in Figure **??**. Note that in order to main the correct ranking of the read IDs, the smallest suffix $ will be appended by the first read in T for sorting. Thus, all the suffixes starting with $ are sorted by the reads following the $. In addition, for any suffix starting with $, their read IDs are determined by the read following the $.

Once the BWT and the auxiliary data structures are constructed, backward search is conducted for any query read, starting from the ending character. If some reads in T have a prefix match with the query and the match size is above a given threshold, the backward search will return a range containing those reads. Using the corresponding RID array, we can obtain their read IDs. Note that bidirectional search algorithm exists [142]. In our work, we reverse the query in order to find the overlap formed by the prefix of the query and suffix of a read in the constructed BWT. The pseudocode of the backward search can be found in Algorithm 1. The input parameters are the BWT, Occ, C, the query read r, the overlap threshold τ, and the read ID array RID. The outputs

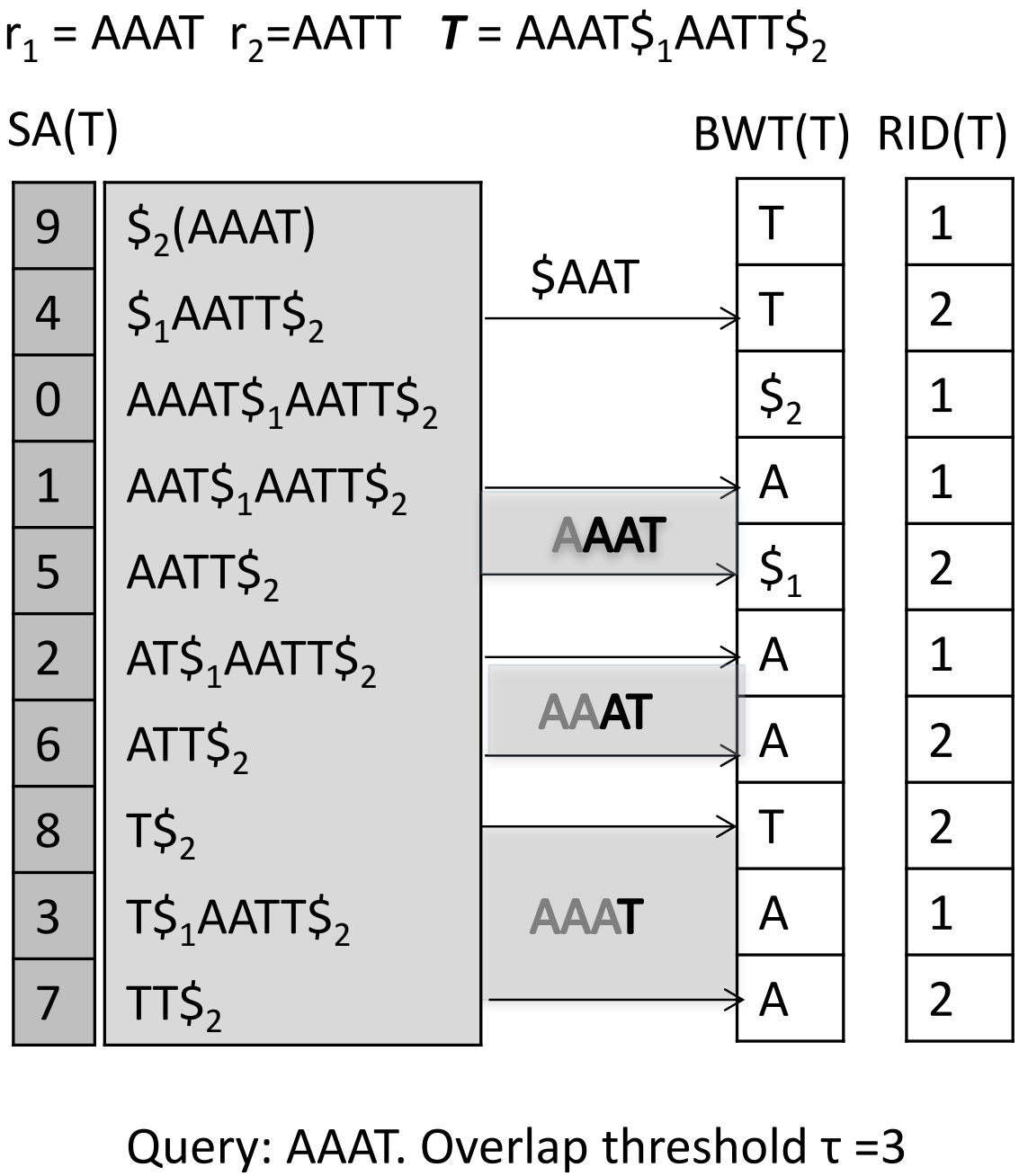

Figure 2.2: The visualization of the output of each step of the backward search for a query sequence AAAT. The SA and the corresponding suffixes are grey because they are not actually used in the search. The computed range for each iteration is highlighted using gray box encompassed by arrows. When τ is 3, the search should return r_2 as it forms an overlap of size 3 with AAAT.

are reads whose prefix match the suffix of r with size $\geq \tau$. An example of the backward search following the pseudocode is provided in Figure 2.2.

Algorithm 1 Overlap detection using BWT's backward search

Function: $overlap((BWT(T), C(\Sigma), Occ(\Sigma, BWT), r, \tau, RID(T))$
Input: T: input text; r: query read; C and Occ: the auxiliary structures; τ: the overlap threshold; RID: read ID array
Output: Reads whose prefix match the suffix of r with size $\geq \tau$

```
1:  i ← |r|
2:  c ← r[i]
3:  start ← C[c] + 1
4:  end ← C[c+1]            ▷ character c+1 is the next character in alphabetically sorted alphabet
5:  while start ≤ end and i ≥ 2 do
6:      c ← r[i − 1]
7:      start ← C[c] + Occ[c, start − 1] + 1
8:      end ← C[c] + Occ[c, end]
9:      i − −
10:     if |r| − i ≥ τ then
11:         start ← C['$'] + Occ['$', start − 1] + 1
12:         end ← C['$'] + Occ['$', end]
13:         if end ≥ start then
14:             output reads with IDs between RID[start] and RID[end]
15:         end if
16:     end if
17: end while
```

2.2.3.1 Unique implementation strategies

Although there are available implementations of the BWT-based overlap detection, ours differs from the existing ones in the following aspects. The first difference is the storage of the read ID information. For a constructed BWT and a query, the output of the backward search is the set of reads (i.e., their IDs) that form overlaps with the query. Theoretically, different reads can be distinguished by appending unique delimiters at the end of each read. Some existing implementations save read IDs for each suffix. For example, SGA assembler [142] saves read ID information along with the suffix starting position in the generalized suffix array $SA(T)$ [142]. In our implementation,

we use '$' as the delimiter for all reads. The read ID array *RID* is created only for suffixes starting with '$'. This works because the backward search algorithm needs to retrieve the read ID in the final step, where the character to search is '$'. This modification significantly reduced the memory usage by reducing the size of *RID* from $|T|$ integers to n (number of reads) integers, where T is roughly the product of n and the read size.

2.2.4 Iterative search

Overlap detection will be iteratively applied to recruit reads sequenced from targeted viruses. Let R_0 be the set of seed reads that can be mapped to given reference sequences (i.e., seed read set). First, the $BWT(T)$ for T is built. The seed reads in R_0 are used as queries to the $BWT(T)$. Then newly identified reads that overlap with the seed reads will be used as new queries to the BWT. The iterations will continue until no new reads can be retrieved. Its pseudocode is described in 2.

Algorithm 2 Default mode: create BWT for all reads

Input: seed read set R_0, the input text T, the overlap threshold τ

Output: Reads that are sequenced from the targeted viruses

output $\leftarrow R_0$

$R \leftarrow R_0$

Create BWT and RID for T: $BWT(T)$, $RID(T)$

while R not empty **do**

    Backward search on $BWT(T)$ to find all reads that overlap with reads in R

    Save them to set R'

    output $\leftarrow$ output $\cup R'$

    $R \leftarrow R'$

end while

return output

2.2.4.1 Running time and memory usage

In the above pipeline, once the BWT is constructed, the suffix array will be deleted. The running time of suffix array and BWT construction is linear to $|T|$. The memory usage of BWT is the product of $|T|$ and the size of each character and thus is linear to $|T|$. The memory usage of the *RID* is the product of n and the size of saving a read ID.

When creating BWT for all reads becomes too expensive, our program supports distributed construction of the BWT and FM-index for large input. Specifically, the program can automatically partition input data into multiple smaller files. BWT is then constructed for each divided data set. The read overlap detection can be run in parallel for each BWT. The identified reads are combined and used as the query for the next iteration of read recruitment. In this case, the largest memory

footprint is determined by the size of each divided read set. By default, the number of partitions is five. This number can be modified by users.

2.2.5 Strain-level assembly

The final outputs of our program are assemblies of viral strains. All recruited reads will be used as input to assembly programs. As our program has a modular structure, this step can be executed by any assembly tool chosen by the users. By default, we include in the package our in-house developed tool PEHaplo [26] for viral haplotype reconstruction. PEHaplo does not require any reference sequences and conducts strain-level assembly using paired-end reads. For the input paired-end reads, PEHaplo constructs a paired-end overlap graph, which augmented standard overlap graphs by adding edges connecting nodes that can form ends of read pairs. Then, a greedy path finding algorithm is applied to search for the paths with the best supports from paired-end reads, where the supports are quantified by the number of contained read pairs and also their distances. The detailed algorithm and implementation of PEHaplo are described in Chapter 3.

2.3 Results and discussion

We have developed a modular structured tool named TAR-VIR for reconstructing viral haplotypes from metagenomics data. The final outputs of this tool are assembled viral contigs corresponding to different strains. We focus on evaluating the performance of the read recruiting stage and also its impact on the final assembly.

Before presenting the results of TAR-VIR for different sets of input data, we first show the results of the longest common substring (LCS) size computation for different microbes.

2.3.1 Sizes of common regions between human viruses and other microbial species

To evaluate the similarity between different microbes, we calculated the sizes of LCSs between human viruses and other microbial genomes. As bacteria infect humans as well as viruses, the sizes of LCSs between human viruses, human vs. non-human viruses, and human viruses vs. bacteria were calculated. The virus reference genomes were downloaded from NCBI Viruses (https://www.ncbi.nlm.nih.gov/genome/viruses/). To date (June 2018), there are in total 7,456 complete viral genomes, of which 481 have human as the natural host (denote as human viruses). The human bacterial reference genomes were downloaded from Human Microbiome Project (HMP) on NCBI. In total, 2,314 bacterial reference genomes were downloaded.

As there is a large number of microbial species available, we conducted LCS search for available microbial genomes by constructing generalized suffix array and the corresponding longest common prefix (LCP) array [50]. *First*, we build a generalized SA [121]. *Then*, the LCP array, which contains LCPs between each two adjacent suffixes, can be calculated in linear time [61, 60]. By definition, the LCS for each two sequences is the maximum LCP between all pairs of suffixes from the two sequences. The following lemma is employed in order to avoid checking all the LCP values between two sequences. For the suffix starting at $SA[i]$, the LCP between $SA[i]$ and $SA[j]$ ($j > i$) is no less than the LCP between $SA[i]$ and $SA[k]$ if $k > j$. With this property and a user-defined LCS cutoff, the LCP calculation between $SA[i]$ and all other suffixes after i can be calculated in constant time. The overall time complexity is $O(N)$.

The results of the LCS histograms are shown in Figure 2.3(A-C). One may also examine whether the read recruitment process can incur contamination by using simulated or real sequencing data. However, the empirical studies using real data are limited to the viruses in the samples.

Meanwhile, producing simulated sequencing data for all microbes is not practical. Using suffix-array based LCS computation allows us to obtain a more comprehensive view of the common regions between different microbes.

We also compared the sizes of the LCSs between different microbes with the ones within a viral population. As the characterized haplotypes for different RNA viruses are very limited, instead of computing the LCS using available data, we estimated the LCSs within a quasispecies using a probability model and dynamic programming [26]. With the mutation rate of 3e-5 at each base during virus replication, the probability distribution of LCS length between two HIV strains that are n generations apart were calculated and the distribution of LCS probabilities is shown in Figure 2.3(D).

2.3.2 Exp1: reconstruct the SARS haplotypes using the bat coronavirus as the reference

In this experiment, we mimic the scenario in which SARS-CoV [23] is an emerging virus infecting humans. Our goal is to reconstruct the SARS-CoV haplotypes using other coronaviruses as references. During the breakout of SARS, electron microscope image reveals the crown-like shape of the infectious agent, providing hints to use coronaviruses as references.

To test this, we assume that the bat coronavirus (NC_014470.1) was sequenced and available to use as a reference, although it was actually sequenced after the breakout of SARS.

2.3.2.1 Data properties and evaluation metrics

A viral metagenomic dataset containing Influenza (NC_002023.1), hepatitis C virus (HCV, NC_004102.1), and 5 SARS-CoV haplotypes, was simulated. The SARS-CoV haplotypes were created from the

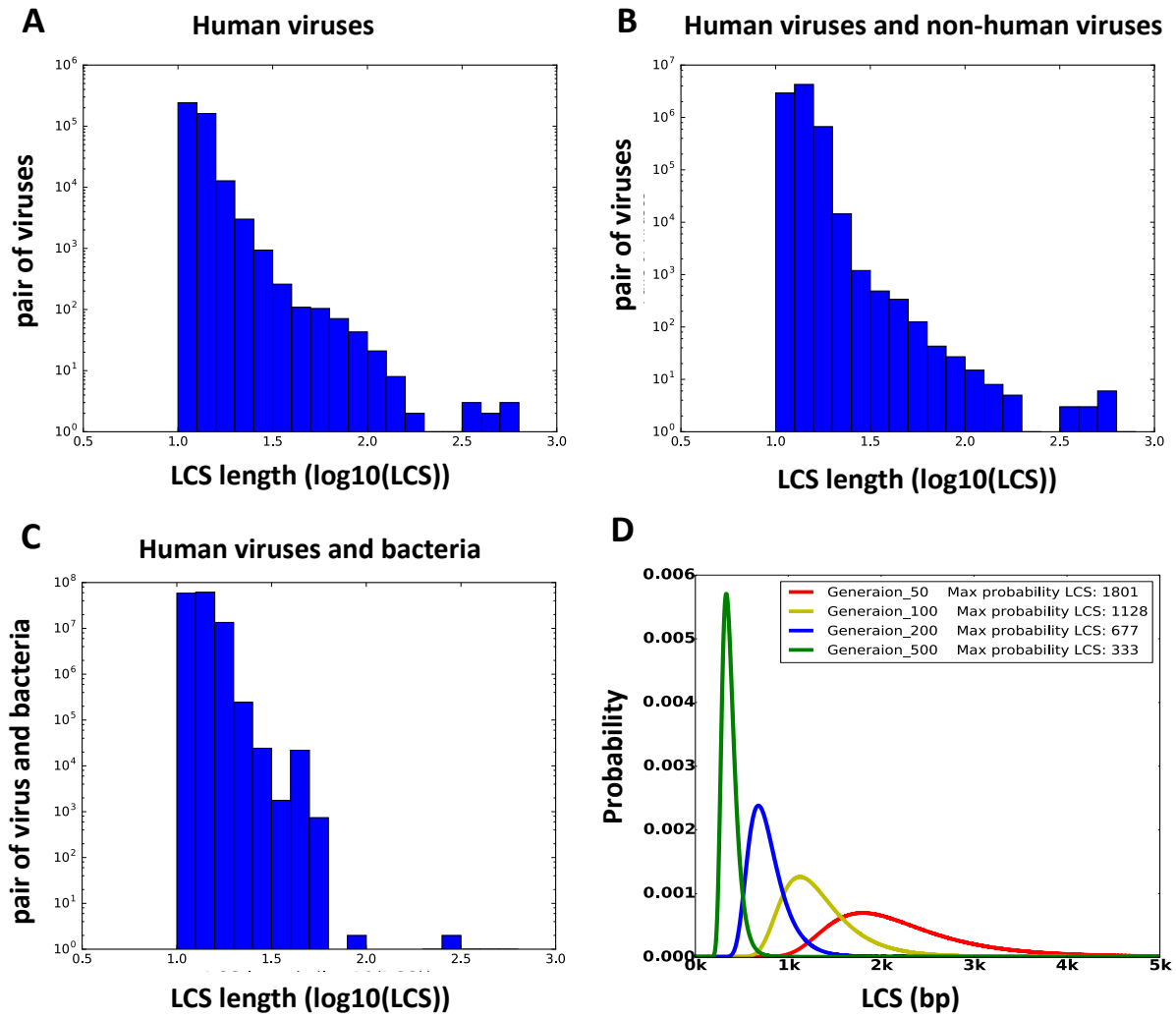

Figure 2.3: Histogram of the LCS sizes between human viruses (A), between human viruses and non-human viruses (B), and between human viruses and bacteria (C). The x-axis is the *log*10 of the LCS length. The y-axis is the number of pairs within the given range of LCS size. Only LCSs that are longer than 10 bp are presented. (D) Probability distribution for LCSs between two simulated HIV strains that are 50, 100, 200, and 500 generations apart. The *x*-axis is the length of LCS, with a range from 0 to 10,000 bp. The *y*-axis is the corresponding probabilities for those LCS sizes.

SARS-CoV reference (NC_004718.3) genome by mutating bases at randomly selected locations. The sequence similarity between any two haplotypes is above 96%. The abundance of each haplotype is calculated based on a power law equation [9]. The total sequencing depths for the 5 SARS-CoV haplotypes are 1000-x, with 438-x, 219-x, 146-x, 109-x, and 88-x for each haplotype, respectively. The sequencing depths for Influenza and HBV are 700-x and 300-x, respectively. All the data sets were simulated by ART-illumina [55] as error-containing MiSeq paired-end reads, with the read length of 250 bp, the average insert size of 600 bp, and the standard deviation of 150 bp. In total, there are 173,703 simulated reads, of which 119,002 reads are from the five SARS-CoV haplotypes.

With the available bat coronavirus as the reference, the simulated reads were aligned with both Bowtie2 [71] and BWA [79]. We then applied the overlap extension component of TAR-VIR to isolate and enrich SARS-CoV reads, and assembled them with *de novo* assembly tools. Both the read recruitment and the assembly performance were evaluated. For the simulated data, the ground truth of the originating haplotype and position of each read is known. Thus read recruiting performance can be evaluated using the reads' positions and originating haplotypes. In summary, we examine how many reads are correctly recruited for each haplotype and report the haplotype coverage and depth.

The assembly performance was evaluated using the known genomes of the 5 SARS-CoV haplotypes and MetaQuast [100]. Similar to other works, we quantified the assembly continuity, completeness, and accuracy in terms of number of contigs, N50, genome coverage, and mismatch rate. N50 is defined as the maximal length so that all contigs above this length contain at least 50% of all the contig bases. Genome coverage is the percentage of the five haplotypes' genomes being aligned by at least one contig. Mismatch rate is the percentage of mismatches between the aligned contigs and the references. In all cases, contigs of at least 500 bp are aligned to the viral refer-

ence sequences for evaluation. The assembly results were also benchmarked with other popular assembly tools SGA [142], SPAdes [8], and SAVAGE [6].

2.3.2.2 Performance of read recruitment

We applied both Bowtie and BWA in the read mapping stage. By adjusting the scoring function related parameters, we constructed different sets of seed reads that can be aligned to the references with different approximate match constraints. For each seed set, the recruited reads generated by TAR-VIR are recorded. Table 2.1 compares the aligned and recruited reads for each SARS-CoV haplotype. Besides approximate match rates, we also considered local and "glocal" alignment mode, where the glocal mode requires the end-to-end alignment of the read against the reference. Using local alignment mode for read mapping can usually produce a larger seed set. However, it is possible that some of the locally aligned reads are not sequenced from the underlying haplotypes. In Table 2.1, we used local alignment mode for BWA and glocal model for Bowtie. Thus, the seed sets constructed by BWA is larger than Bowtie.

Even with the least stringent threshold, the aligned reads have lower genome coverage than recruited reads, which is expected because SARS-CoV does not share genome-scale high similarity with the bat coronavirus (Figure 2.4 (C)). In particular, with the parameter "-B 1", BWA can align slightly more reads than what Bowtie2 can recruit with the parameter "L, 0, -0.9" ((52,688 vs. 52,377). The recruited reads (52,377), however, cover 20%-30% more genomes for the five haplotypes. This indicates that alignment-based methods tend to identify reads sequenced from highly similar regions between the target and the reference viruses, while the recruitment method is more likely to obtain reads from the whole genome of the target viruses. Worth noting is all the recruited reads are from SARS-CoV (no contamination from Influenza and HBV).

Figure 2.4(A) and Figure 2.4(B) compared the genome coverage of seed reads and recruited

Table 2.1: Read recruitment results by using seed sets constructed with Bowtie2 and BWA. The "Alignment" section contains results for aligned reads. The "Recruitment" section contains results for recruited reads by TAR-VIR using the aligned reads. For each row, the aligned reads in "Alignment" section are the seed set for the recruited reads in "Recruitment" section. For Bowtie2, the "–score-min" parameter was set to allow different alignment error rates corresponding to 5%, 10%, 15%, and 20%, respectively. For BWA, "-A" is fixed as its default value 1. "-B" was modified to allow different error rate similar to Bowtie2. "Number" is the number of aligned or recruited reads. "Depth" is the average sequencing coverage. "Coverage" is the percentage of genome covering by at least one read. h1 to h5 represent the five SARS-CoV haplotypes.

Bowtie2	Alignment										
	Number	Depth					Coverage				
		h1	h2	h3	h4	h5	h1	h2	h3	h4	h5
L,0,-0.3	55	0.13	0.01	0.06	0.15	0.12	0.01	0.01	0.01	0.01	0.01
L,0,-0.6	925	3.6	1.5	0.9	0.9	0.9	0.07	0.06	0.05	0.07	0.08
L,0,-0.9	8,154	32	14	9	8	7	0.31	0.31	0.27	0.27	0.3
L,0,-1.2	13,221	49	24	15	13	10	0.43	0.45	0.42	0.44	0.43
	Recruitment										
	Number	Depth					Coverage				
		h1	h2	h3	h4	h5	h1	h2	h3	h4	h5
L,0,-0.3	45,504	182	89	59	41	11	1.0	1.0	1.0	1.0	0.37
L,0,-0.6	46,576	183	90	60	42	18	1.0	1.0	1.0	0.96	0.55
L,0,-0.9	**52,337**	198	96	63	46	37	1.0	1.0	1.0	0.98	0.99
L,0,-1.2	55,485	182	89	59	41	39	1.0	1.0	1.0	1.0	0.99
BWA	Alignment										
	Number	Depth					Coverage				
		h1	h2	h3	h4	h5	h1	h2	h3	h4	h5
B:8	24,585	89	46	28	20	18	0.4	0.37	0.33	0.31	0.34
B:4	41,564	152	78	50	37	32	0.63	0.57	0.56	0.57	0.53
B:2	51,995	195	94	63	46	39	0.79	0.78	0.77	0.70	0.74
B:1	**52,688**	199	94	64	47	39	0.81	0.78	0.80	0.71	0.76
	Recruitment										
	Number	Depth					Coverage				
		h1	h2	h3	h4	h5	h1	h2	h3	h4	h5
B:8	62,609	235	117	75	55	44	1.0	1.0	1.0	1.0	0.99
B:4	72,901	270	135	89	65	53	1.0	1.0	1.0	1.0	1.0
B:2	79,755	299	146	97	71	58	1.0	1.0	1.0	1.0	1.0
B:1	78,540	294	143	96	70	57	1.0	1.0	1.0	1.0	1.0

reads. Directly aligning the reads to the bat coronavirus covers only a small proportion of the whole genome (Figure 2.4(A)), leading to incomplete assembly. Using these aligned reads as seeds, TAR-

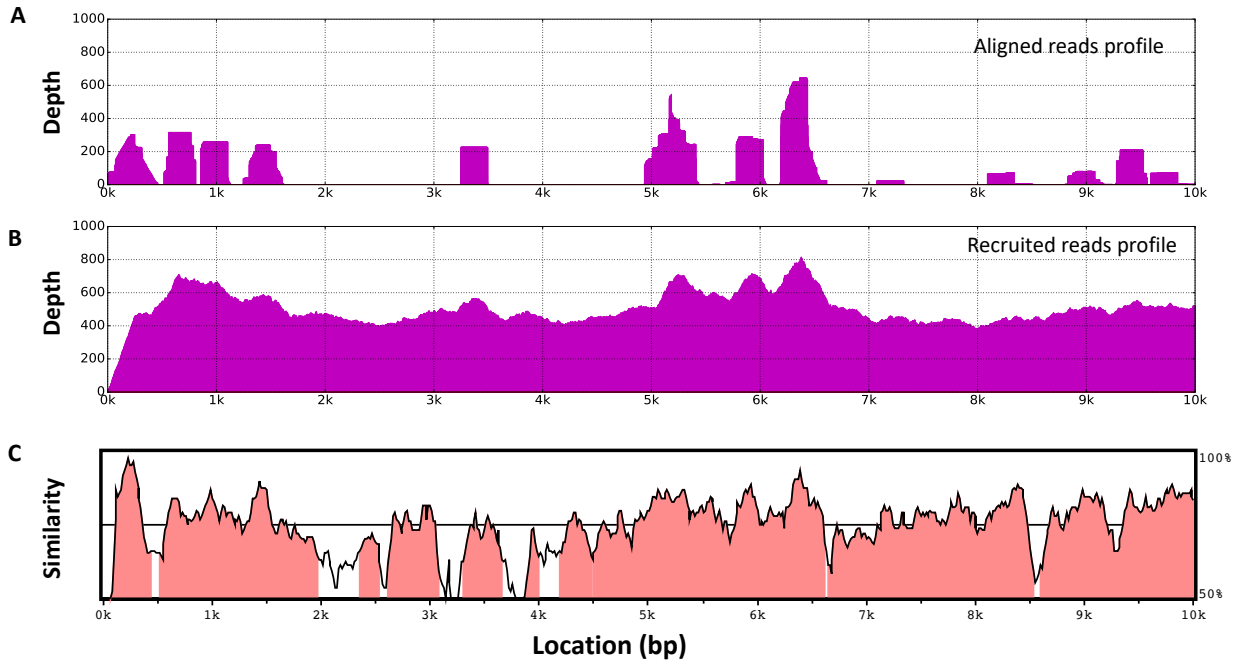

Figure 2.4: Enriching SARS-CoV reads using the bat coronavirus genome as the reference. (A) and (B) show the aligned and recruited reads profile. The dataset was aligned by BWA with the default parameter ("-B 4, -A 1"). BWA is chosen to include more locally aligned reads in (A). The reads were recruited using the overlap cutoff of 150 bp. (C) displays the sequence identity between SARS-Cov and the bat coronavirus. The profile was generated using VISTA [45].

VIR is able to recruit many more reads that nearly cover the whole genome of SARS-Cov, as shown in Figure 2.4(B).

Table 2.1 also shows that the numbers of recruited reads do not heavily rely on the number of the seed reads. Even when the seed set is small (e.g. the seed sets constructed using Bowtie2), many new reads can be recruited during each iteration. After multiple iterations, the final set of recruited reads can be significantly larger than the seed set, bounded by the sequencing depth of the haplotypes. On the other hand, if the seed sets contain many reads from non-relevant species, the final set of recruited reads could even include all the reads from the input, which makes the read recruiting useless. Because of this, we prefer to construct the seed set using glocal mode to ensure high quality.

Recruited reads lead to better assembly performance

Both the aligned and recruited reads were assembled with *de novo* assembly tools. PEHaplo is the default assembly component in TAR-VIR. As TAR-VIR has a modular structure, other *de novo* assembly tools including SGA, SPAdes, and SAVAGE are also used to replace PEHaplo for haplotye reconstruction. SPAdes was run with –meta option, which is same as metaSPAdes [108]. As the aligned reads cover at most 80% of the genomes even with the least stringent alignment threshold, it is not proper to apply the conventional reference-based assembly methods for this data set.

The complete *de novo* assembly results using both aligned and recruited reads are presented in Supplementary Table S1. Part of the results are shown in Table 2.2 due to space limiation. For all assembly tools, using recruited reads produces better results: longer contigs and higher genome coverage. Significantly, this is not simply due to the increased number of reads. For example, as shown in Table 2.2, the reads recruited by Bowtie2 with parameter "L, 0, -0.9" is less than BWA-aligned reads when B is 1 (52,337 vs. 52,668). But the recruited reads produce contigs at least ten times longer than the aligned reads, and twice higher in the genome coverage. By comparing the assembly performance of all tested tools on the recruited reads, our assembly component PEHaplo consistently has higher N50 and genome coverage than others. Overall, PEHaplo and SGA perform better than the other two assembly tools.

PRICE applies extension-based strategies for contig assembly. Using the seed reads as initial contigs, PRICE can be readily used to perform targeted viral assembly from metagenomic data. Therefore, the results of TAR-VIR was also benchmarked with PRICE's results, as shown in Table 2.3. PRICE produced one long contig (similar N50 to ours) for the most abundant haplotype. Thus, its genome coverage is only about 20%.

Table 2.2: Assembly results on SARS-CoV aligned and recruited metagenomic data. The default assembly tool in TAR-VIR is PEHaplo. The definitions of the metrics can be found in Section 2.3.2.1.

Aligned	Tool	# Contigs	N50	Genomes covered (%)	Mismatch rate (%)
Bowtie2 L,0,-0.9	TAR-VIR	58	505	19.7	0.02
	SGA	56	505	20.1	0.03
	SPAdes	34	569	12.9	0.16
	SAVAGE	54	455	17.5	0.0
Recruited	Tool	# Contigs	N50	Genomes Covered (%)	Mismatch rate (%)
Bowtie2 L,0,-0.9	TAR-VIR	7	29,676	98.9	0.0
	SGA	13	26,729	98.9	0.0
	SPAdes	14	15,882	92.1	0.51
	SAVAGE	22	12,445	97.0	0.0
Aligned	Tool	# Contigs	N50	Genomes covered (%)	Mismatch rate (%)
BWA B:1	TAR-VIR	84	1,192	55.1	0.0
	SGA	85	1,027	56.5	0.0
	SPAdes	67	1,012	44.6	0.12
	SAVAGE	68	669	32.3	0.0
Recruited	Tool	# Contigs	N50	Genomes Covered (%)	Mismatch rate (%)
BWA B:1	TAR-VIR	6	29,706	99.5	0.0
	SGA	18	12,638	99.5	0.0
	SPAdes	21	10,353	89.2	0.39
	SAVAGE	56	5,140	89.3	0.0

Table 2.3: Assembly results on SARS-CoV metagenomic data for TAR-VIR and PRICE.

	Tool	# Contigs	N50	Genomes Covered (%)	Mismatch (%)
Bowtie2 L,0,-0.9	TAR-VIR	7	29,676	98.9	0.0
	PRICE	1	29,749	20.0	1.7
BWA B:1	TAR-VIR	6	29,706	99.5	0.0
	PRICE	1	29,750	20.0	1.66

2.3.3 Exp2: characterizing hepatitis viruses from the human plasma data

In this experiment, TAR-VIR is tested on a real metagenomic data set, which was sequenced from the plasma of 19 antiretroviral-treated HIV patients (SRR2083204) [80]. The samples were pre-amplified by random RT-PCR amplification (RA) for both viral RNAs and DNAs and then

sequenced by Illumina Miseq, producing about 23 million reads. All these samples contain low levels of HIV because of the antiretroviral treatment. But it may contain other human pathogens. In our study, we focused on identifying hepatitis viruses. Although our pipeline is designed to tackle the challenges of characterizing RNA viral quasispecies, we also include in the references DNA hepatitis viruses such as HBV.

2.3.3.1 Preprocessing

The raw data set contains reads that come from varied sources: human, bacteria, phages, etc. The reads of the target viruses comprise less than 30% of the entire data set. Since the primary focus is human viruses, removing those reads from the host (human), bacteria, and phages is ideal before pathogen detection. Following canonical quality control and trimming, we used bamtagger [124] to remove human reads, and Bowtie2 to remove reads from bacteria and phage by aligning reads against their reference genomes. The remaining reads were corrected by error correction tool Karect [2]. After preprocessing, 8,145,722 reads were left.

2.3.3.2 Recruited reads by TAR-VIR can improve the performance of *de novo* assembly

In the first step, we conducted read mapping to obtain the seed reads. Both BWA and Bowtie2 could be used. However, although BWA aligned more reads, many reads yielded only short local alignments and are unlikely to be sequenced from the target viruses. Using these reads as seeds tend to cause contamination during the read recruitment stage. For example, when BWA ("-B 8, -A 1") was used to generate the seed set, roughly 3.5 million reads were recruited, while a portion of them can be aligned to other genomes (such as phages). Although BWA's output can be processed to remove local alignments, the seed set can be more reliably produced using Bowtie2's output. Therefore, Bowtie2 was chosen as the aligner for all real data experiments.

We downloaded the reference genomes of HBV (NC_003977.2), multiple genotypes of HCV (NC_009827.1, NC_009823.1, NC_009825.1, NC_030791.1, NC_004102.1, NC_009826.1, NC_009824.1), and human pegivirus (HPgV, NC_001710.1) from the viral genome database of NCBI. The pre-processed reads were then aligned to the references under mismatch rates of 5%, 10%, 15%, and 20%, respectively. These initially aligned reads were used as the seed read sets. Although there are multiple genotypes for HCV, only genotype 1 has a decent amount of aligned reads. Other genotypes have less than 50 reads mapped. Thus, to produce a reliable evaluation of the assembly results, only the results of HCV genotype 1 were used. The numbers of reads before and after read recruiting are shown in Table 2.4.

Table 2.4: Overlap extension results using different seed set R_0. '#' represents 'number'. The shaded regions in this table and Table 2.5 highlight the case where less recruited reads can produce better assembly results than aligned reads only.

Align mismatch	Seed #	Recruited #	Align mismatch	Seed #	Recruited #
5%	21,925	200,650	10%	67,973	222,065
15%	162,454	263,029	20%	294,448	340,705

As this is a real metagenomic data set without known ground truth of the viral haplotypes, the evaluation metrics for read recruiting are different from the simulation data set. We cannot evaluate whether every recruited read is correct because its originating location is unknown. Thus, instead of evaluating the depth and genome coverage for each haplotype, we focus on evaluating whether using recruited reads can improve the performance of genome assembly.

Therefore, both the aligned reads and the recruited reads from TAR-VIR were assembled by *de novo* assembly tools, and the results were compared in Table 2.5. The assembly results demonstrate that the reads recruited by TAR-VIR usually improve the assembly results by producing longer contigs and higher genome coverage for its PEHaplo, SGA, and SPAdes. The improvement is not

simply due to the increased number of reads after the recruitment stage. For example, according to Table 2.4, by using 15% mismatch rate, the recruited reads are less than the aligned reads under 20% mismatch rate (263,029 vs. 294,448). However, the assembly results using the recruited reads are better than or comparable to the results using the aligned reads for all the assembly tools. Among the four assemblers used, PEHaplo of TAR-VIR and SPAdes produced good results with large N50 and high genome coverage. SGA generated larger number of contigs with low N50 value. While we have tried the best parameters for SAVAGE with our empirical experience, its results are not consistent with the other three tools. Better parameters may exist for SAVAGE to produce better results. However, the long-running time and high memory usage of this tool made continuing to tune the parameters difficult.

Table 2.5: Assembly evaluation results on aligned and recruited reads using the genomes of HBV, HCV, and HPgV as references. 'cov.' is the abbreviation for 'coverage'. The default assembly component in TAR-VIR is PEHaplo.

		Bowtie2 aligned			Bowtie2 Recruited		
Align	Tool	Contig #	N50	Genome cov. (%)	Contig #	N50	Genome cov. (%)
5%	TAR-VIR	11	920	27.3	97	3,643	82.3
	SGA	14	645	26.8	63	675	68.4
	SPAdes	5	1,177	27.6	15	3,636	79.6
	SAVAGE	13	698	21.6	49	806	40.4
10%	TAR-VIR	61	794	67.4	31	2,635	84.0
	SGA	26	663	56.4	72	706	69.5
	SPAdes	15	1,251	65.4	19	3,373	79.3
	SAVAGE	30	631	40.6	32	915	26.5
15%	TAR-VIR	97	939	80.9	14	3,579	83.1
	SGA	56	617	57.7	74	722	70.2
	SPAdes	20	1,689	77.6	16	3,986	81.0
	SAVAGE	32	639	29.9	24	999	27.6
20%	TAR-VIR	38	1,852	84.5	77	5,678	86.4
	SGA	78	661	59.5	374	537	64.5
	SPAdes	23	2,710	83.8	10	4,830	84.6
	SAVAGE	19	671	19.1	15	823	5.5

2.3.3.3 Comparison with reference-based and extension-based assembly methods

With the reference genomes available, reference-based tools can be applied for viral metagenomic data analysis. Therefore, we also benchmarked TAR-VIR with reference-based haplotype reconstruction tools including Haploclique [147], drVM [86], and ViQuas [59]. VirusTap [162] can also conduct reads classification and then assembly. While we were planning to compare TAR-VIR with VirusTap, a large data set could not be uploaded to the website-based VirusTap. In addition, about 3,000 jobs were waiting at the website. Therefore, the results from VirusTap could not be reported.

The reads aligned with the mismatch rate of 15% were used as input for Haploclique and ViQuas. For drVM, the reference genomes were built from human viruses, and it ran on the raw fastq files (with simple quality control and trimming) dumped from SRA file with default parameters. The seed read set of TAR-VIR were also the reads mapped with 15% mismatch rate. The assembly results are shown in Table 2.6.

Table 2.6: Assembly results comparison with reference- and extension-based methods.

Tool	Contig #	N50	Genome cov. (%)	Tool	Contig #	N50	Genome cov. (%)
TAR-VIR	14	3,579	83.1	ViQuas	396	9,646	100.0
Haploclique	50,419	304	71.1	drVM	413	829	81.9

The results show that TAR-VIR performs better than Haploclique and drVM by producing fewer but longer contigs with higher genome coverage. With the complete and also the likely "true" virus genomes as the reference, ViQuas has produced near-complete genomes. However, it produces almost 400 contigs with similar lengths (full genomes), indicating a high probability of overestimation of the haplotypes. Since the ground truth of the actual number of haplotypes in this data set is unknown, we intended to test this hypothesis using a dataset with known haplotypes. Therefore, we tested ViQuas on the SARS-CoV simulated data set with 5 haplotypes. It reported

113 contigs, each covering 99.98% of the genome with high mismatch rate ($> 9.0\%$). Thus, the long contigs produced by ViQuas are not likely the true haplotypes.

Similar to SARS-CoV data, we also benchmarked our results with the extension-based tool PRICE. The initial contigs of PRICE were also the reads mapped with 15% mismatch rate. PRICE generated 164 contigs, with a N50 value of 791, and genome coverage of 87.3%. PRICE's results have a slightly larger genome coverage but a much smaller N50 value comparing to TAR-VIR.

2.3.3.4 Assembling the whole data set directly

As SGA and SPAdes are highly efficient and have been used by various virus analysis pipelines, it may be possible to directly apply them to all the preprocessed reads for recovering the three viruses ((HBV, HCV genotype 1, and HPgV). Thus, we applied SGA and SPAdes to the preprocessed reads. The assembled contigs were compared with the reference genomes of the three viruses. SGA took about 1 hour to finish. It generated 2,659 contigs, from which 123 contigs can be aligned to the three viruses with the similarity threshold of 90%. The 123 contigs can cover 42.36% of the reference genomes. SPAdes failed to report the results within 24 hours. The results from SGA verified that although the preprocessed data set contains all the reads from the target viruses, the sheer data size and the low proportion of the three viruses make generic assembly difficult. Meanwhile, assembling a large data set consumes significant computing resources.

2.3.4 Identifying viruses containing target genes

In some situations, the researchers are only interested in the viral genomes containing a partial or complete gene. In these cases, it is difficult for existing reference-based virus identification tools to construct the whole viral genome. Here, we demonstrate that with the overlap extension method, the most of a genome can be built from a partial gene reference.

In this experiment, we show that with a non-complete gene sequence of length 1,073 bp for HPgV as reference, most of the genome can be assembled. The reference sequence (Sequence name: 10MYKJ037) was downloaded from Virus Pathogen Database and Analysis Resource (ViPR) [117], which is a partial coding DNA sequence (CDS) of HPgV isolated from Malaysia in 2010. The total length of HPgV genome is 9,392 bp. From the results of our previous experiments, overlap extension from aligned reads under mismatch rate of 15% was able to recruit adequate reads while keeping away unreliable reads as seeds. Therefore, we aligned the raw reads to this CDS reference by allowing mismatch rate of 15%, from which 19,714 reads were aligned. ViQuas and drVM were used to assemble the aligned reads. However, ViQuas could only produce contigs similar to the short CDS sequence. In addition to provided CDS reference, drVM also downloaded references from Internet. It correctly recognized the HPgV but failed to produce any contig. The results confirm that reference-based methods do not apply in this case. With overlap cutoff of 150 bp, 118,339 reads were recruited from the overlap extension step. They were then assembled by PEHaplo of TAR-VIR, SGA, SPAdes, and SAVAGE, as shown in Table 2.7.

Table 2.7: Assembly results on recruited reads with a partial CDS sequence for HPgV as reference.

Tool	Contig #	N50	HPgV cov. (%)	Tool	Contig #	N50	HPgV cov. (%)
TAR-VIR	5	7,959	86.0	SPAdes	6	8,957	94.0
SGA	41	591	49.0	SAVAGE	35	580	49.5

While the length of reference strain being only 11.42% of the whole HPgV genome, the contigs assembled from recruited reads by TAR-VIR and SPAdes are able to cover the nearly complete genome. The results reveal that even with a gene/CDS sequence as reference, sufficient reads can still be collected to construct the virus at the whole genome level. As there is only one target virus, SPAdes produced the best results. Applying SPAdes to the whole human plasma data set failed

to finish on the cluster after 24 hours, but by using recruited reads, SPAdes can produce better assemblies with the minimum amount of resources.

2.3.5 Computational time and memory usage

We evaluated the time and memory usage of TAR-VIR on the real human plasma data. After preprocessing, 8,145,722 reads were left. The data size is 2.9 GB, and the total length of the sequences are 2,447,741,491 bp. To reduce memory usage, the raw data was split into 5 parts, with 5 BWTs being built for the whole data. The splitting process is embedded in our program, and the number of segments can be set by users. For each partition, the file sizes for the BWT, Occ array, and the read ID array are 490M, 200M, and 13M, respectively. The total size of built indexes is 3.5 GB. The detailed time and memory usage for the overlap extension is shown in Table 2.8 below. A user can load each partition separately to reduce the memory usage. In that case, the memory usage is about the size of each partition. In addition, we may further reduce the memory usage of the recruiting process by applying more compact implementation of the BWT [18].

Table 2.8: Time and memory usage for overlap extension and assembly on viral metagenomic data from human plasam. The *de novo* assembly time and memory usage were evaluated on recruited reads based on mismatch rate from 5% to 20%.

		Time	Memory(GB)
Overlap extension	Building index	127m	2.4
	Recruitment	$<$ 20m	$\sim$ 3.5
De novo Assembly	TAR-VIR	5m - 20m	2 - 4
	SGA	$<$ 10m	$<$ 1
	SPAdes	$<$ 10m	$<$ 1
	SAVAGE	2h - 72h	64
	PRICE	$\sim$2h	$\sim$5.2
Reference-based assembly	Haploclique	33h	5
	ViQuas	$>$ 72h	$<$ 4
	drVM	$<$ 30m	$<$ 1

All the experiments were tested on an MSU HPCC CentOS 6.8 node with Two 2.4Ghz 14-core

Intel Xeon E5-2680v4 CPUs and 128GB memory. We used 4 threads for the assembly component of TAR-VIR, 8 threads for SGA, 16 threads for SAVAGE, 8 threads for PRICE, 1 thread for Haploclique and ViQuas, and 2 threads for drVM.

2.4 Conclusions

In this Chapter, we presented a novel pipeline for viral reads classification and strain-level assembly from viral metagenomic data named TAR-VIR. When a virus in a metagenomic dataset is only remotely related to a characterized virus in public databases, our pipeline can be applied to first classify the reads belonging to these viruses and then conduct strain-level assembly. Or if a user is interested in detecting a virus that contains a given gene, our method can be employed to recover the whole genome of the gene-containing virus.

We also made contributions by conducting careful analysis of the common region sizes between and within viral quasispecies. These analyses laid the foundation for using overlap detection to classify reads of the same quasispecies without introducing contamination. Our unique implementation of the indexing structure also make our method economical in both memory and CPU usages.

We demonstrated the tool's utilities on a simulated viral metagenomic data containing SARS-Cov and a real viral metagenomic data set sequenced from human plasma. The simulated data enables us to evaluate the performance of read classification to the resolution of each single read. It shows that TAR-VIR can successfully classify enough reads to cover the whole genome. In addition, it produced contigs covering five different haplotypes.

On the human plasma data, we were able to enrich enough reads from the target viruses for downstream assembly even with a small seed read set. With a partial CDS sequence for HPgV as

reference, TAR-VIR was able to produce near complete genome assemblies. The results clearly showed the effectiveness of TAR-VIR. In summary, TAR-VIR provides complementary functions to existing virus detection tools when the quality or complete references are not available.

Chapter 3

De novo haplotype reconstruction in virus quasispecies using paired-end reads

3.1 Introduction

Virus quasispecies describe a population of different but closely related virus with dynamic distribution. Each strain in quasispecies is defined by its haplotype sequence. Nature selection, point mutation, and recombinations can all change the haplotypes and their abundance inside a quasispecies. RNA virus and some DNA virus evolve following the quasispecies model. Commonly known examples include human immunodeficiency virus (HIV-1), the hepatitis C virus (HCV), the foot-and-mouth virus (FMV), etc. Figure 3.1 shows a local multiple alignment of five haplotypes in HIV quasispecies.

As the selection works on a set of sequences rather than one, quasispecies have abilities to escape host immune responses or develop drug resistance. Reconstruction of the viral haplotypes is a fundamental step to characterize the quasispecies, predict viral phenotypes, and finally provide important information for clinical treatment and prevention.

Development of next-generation sequencing technologies sheds light on characterizing the haplotypes and their abundance in quasispecies. If the reference sequences are available, read mapping can be conducted to infer haplotypes. However, due to the high mutation rate, usually quality reference genomes of quasispecies are not available. Thus, there is a need for *de novo* haplotype re-

construction method. A recent review of chosen haplotype assembly tools has shown that *de novo* haplotype recovery is a computationally challenging problem. It shares some common challenges with metagenomic assembly, which aims to assemble short reads into full genomes of member species. The essential idea is to build a graph (e.g. De Bruijn or overlap graph) using reads in the given sample and then recover the genome by walking through a path. In the ideal case, where there are no sequencing errors, there is sufficient coverage, and the genomes differ substantially, it is trivial to identify reads belonging to the same haplotype and stitch them into a long sequence. However, in reality, haplotype construction must handle short error-prone reads, high similarity between different strains, rare mutations, and low abundance of some strains. We briefly discuss how these issues complicate the assembly problem. As we construct overlap graph for haplotypes recovery in this work, we focus on how these issues make *de novo* assembly in overlap graphs difficult.

First, if the common region between any two strains is shorter than the read length, the connection of all reads will lead to different paths, each of which corresponds to the full sequence of a haplotype. However, the common regions between strains can easily exceed the size of a read and thus introduce many shared nodes and bifurcations in the overlap graph. Second, sequencing errors can introduce wrong connections, tips, bubbles and significantly add to the complexity of the graph. Third, alignment-based error correction have difficulty in distinguishing rare mutations from sequencing errors. Finally, reads originating from haplotypes of low abundance tend to have small overlaps and thus only fragments of haplotypes be reconstructed. Indeed, a recent benchmarking [136] demonstrated that the performance of all the tested programs is poor when sequence divergence is low. In addition, these programs failed to recover rare haplotypes. Thus, there is a need for new methods and tools for more accurate haplotype assembly.

There exist a number of generic assembly tools [106, 150, 73, 116, 90, 133, 159], which can be

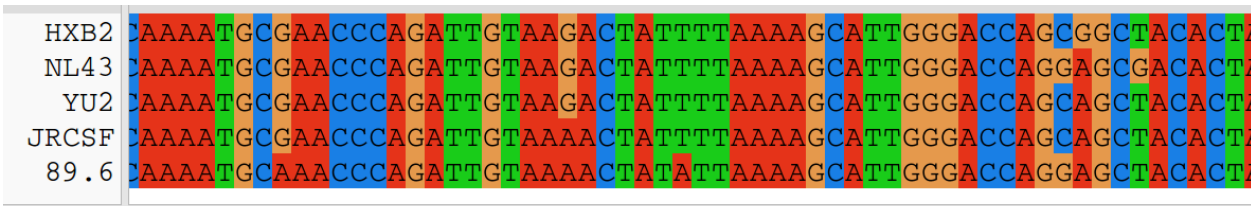

Figure 3.1: Multiple sequence alignment of five HIV-1 haplotypes.

applied for sequence assembly in viral quasispecies data. However, these generic assembly tools are not designed for recovering viral haplotypes in quasispecies. High similarity between different strains, plus sequencing errors, can lead to highly tangled graphs. Applying existing tools either lead to very short contigs or chimeric contigs.

There are a group of tools designed specifically for reconstruction of haplotypes [166, 92, 118, 148, 119, 4, 146, 54]. Some tools focused on estimating local haplotypes. Our goal of this work is global reconstruction, which aims at reconstructing the full-length viral haplotypes in a sample. Compared to local reconstruction, it provides a more complete and accurate characterization of the quansispecies. Some tools such as ShoRAH rely on reference sequences for conducting alignments. For more general application of haplytype reconstruction, we need a *de novo* assembly tool that does not rely on reference sequences.

Of the available haplotype reconstruction programs, there are three highly related ones to ours. They are HaploClique [147], SAVAGE [5], and MLEHaplo [93]. Like PEHaplo, all of them are designed for paired-end reads. HaploClique uses enumeration of maximal cliques as a general approach to clustering NGS paired-end reads. Although PEHaplo is also designed for paired-end reads, the two are highly different in using paired-end reads information. First, HaploClique needs a reference sequence for generating alignment graphs while ours does not rely on reference sequences. Second, HaploClique uses the insert sizes for detection of large indels. However, insert size usually has a distribution and can vary substantially between different fragments. Thus this

information is used more conservatively in PEHaplo. MLEHaplo also explicitly employs paired-end reads for finding top-score paths. Its usage of paired-end information in a De Bruijn graph is highly different from our method based on an overlap graph. HaploClique provides a source of inspiration for SAVAGE [5], which is the first tool for recovering viral haplotypes using overlap graphs. SAVAGE also took advantage of paired-end reads and merge short reads using cliques. The authors benchmarked SAVAGE with other virus assembly tools and showed that SAVAGE outperformed other tools in a comprehensive set of assembly metrics. As we focus on *de novo* assembly tools, we will benchmark PEHaplo against SAVAGE and MLEHaplo.

In this work, we designed and implemented PEHaplo, which assembles virus haplotypes from virus metagenomic sequencing data. Sequence assembly has been an intensive research area and new methods or implementations are emerging quickly. PEHaplo adopted relevant techniques such as error correction and efficient overlap construction. Most importantly, we made contributions to distinguish highly similar haplotypes using paired-end reads. The rationale will be detailed in the Methods Section. We applied PEHaplo to both simulated and real viral quasispecies data and compared the assembly performance with the recently published tools. The experimental results show that PEHaplo can recover viral haplotypes with longer contigs and higher accuracy.

3.2 Methods

In this section, we will first describe the graph models we use for haplotype construction. Then we will present the key idea of applying paired-end reads to distinguish different viral haplotypes. Finally, we show the whole pipeline and detail each main component.

3.2.1 Overlaph graph and paired-end graph

An overlap graph $G(V,E)$ is a weighted directed graph that reflects overlaps between reads. Each node $v \in V$ represents a read. An overlap between two reads is formed if the suffix of a read matches the prefix of another read. Given any two reads r_1, r_2, and an overlap threshold l, if the overlap size between r_1 and r_2 is greater than l, a directed edge is added from the nodes representing r_1 and r_2 in G. The edge weight is the overlap size.

While an overlap graph records the connectivity between reads, we also record the number of paired-end reads between nodes using a paired-end graph PE_G. A paired-end graph is a weighted undirected graph, which has the same node set as the overlap graph. Two nodes are connected by an edge if the corresponding reads form a read pair. In practice, a node in the overlap graph can contain multiple reads after we combine nodes without bifurcations. Thus, multiple read pairs can exist between nodes. The total number of read pairs between two nodes is labeled as the edge weight in PE_G.

In Figure 3.2(B), the edges in overlap graph are shown using solid lines while the edges in the paired-end graph are shown using dashed lines. Nodes a.1 and a.2 form a read pair and thus have an edge of weight 1 in PE_G. Similarly, nodes d.1 and d.2 have an edge of weight 1 because d.1 and d.2 are a read pair.

3.2.2 Mutation Rate for Sequence Replication and Probability of Longest Common Substring (LCS)

Different strains of a virus quasispecies usually share high sequence similarity. But the mutations, insertions or deletions can happen randomly. Instead of sequence similarity, we use longest common substring (LCS) to characterize the similarity between strains, from which strain-level as-

sembly is possible if the LCS is shorter than the paired-end insert size. Here we will first infer the mutation rate accumulation between an initial strain and its nth offspring, then provide a dynamic programming algorithm for calculating the probability of LCS with length m between the initial strain and its offspring.

Mutation rate accumulation for sequence replication Imagine the environment initially with only one virus strain x_0 of length L, mistakes can happen each time when the genome replicates and finally will result in an equilibrium with multiple strains of constant abundances, which is described by the quasispecies theory.

Reproducing mistakes can be insertions, deletions or substitutions. To simplify the problem, we assume only substituions can happen during replication and the virus genome length remains unchanged. Assume the probability of mutation at each position when replicating is constant and independant to each other, denoted as μ, we ask what is mutation probability at each location between x_0 and its nth offspring x_n?

This problem can be solved by dynamic programming. Let $f(i)$ be the probability that $x_n[i]$ is different from $x_0[i]$, and $g(i)$ be the probability that $x_n[i]$ is same to $x_0[i]$. We get

$$\begin{cases} f(i+1) = (1-f(i))\mu + f(i)(1-\mu/3), & i = 1,2,3,...,L \\ g(i+1) = g(i)(1-\mu) + f(i)\mu/3, & i = 1,2,3,...,L \end{cases} \tag{3.1}$$

Note that $f(1) = \mu$ and $g(1) = 1-\mu$, and $f(i)+g(i) = 1$, equations 3.1 can be solved

$$\begin{cases} f(i) = (1-\frac{4\mu}{3})^{i-1}(\mu - \frac{3}{4}) + \frac{3}{4}, & i = 1,2,3,...,L \\ g(i) = (1-\frac{4\mu}{3})^{i-1}(\frac{3}{4} - \mu) + \frac{1}{4}, & i = 1,2,3,...,L \end{cases} \tag{3.2}$$

Probability of LCS For the initial virus strain x_0 of length L, we ask what is the probability of LCS with length m between x_0 and its nth offspring x_n?

Again this problem can be solved by dynamic programming. Define $f(i)$ as the probability that prefix $x_n[1 \quad to \quad i]$ has $LCS \leq m$ with x_0 and $x_n[i]$ mutated, $g(i)$ as the probability that prefix $x_n[1 \quad to \quad i]$ has $LCS \leq m$ with x_0 and $x_n[i]$ not mutated. τ as the probability that $x_n[i]$ is different to $x_0[i]$, which can be calculated using equation 3.1. Now we look at the case when

$$
\begin{aligned}
& m = 2 \\
& f(i+1) = \tau f(i) + \tau g(i) \\
& g(i+1) = (1-\tau) f(i) + (1-\tau)^2 f(i-1) \\
& m = 3 \\
& f(i+1) = \tau f(i) + \tau g(i) \\
& g(i+1) = (1-\tau) f(i) + (1-\tau)^2 f(i-1) + (1-\tau)^3 f(i-2) \\
& \vdots \\
& m \\
& f(i+1) = \tau f(i) + \tau g(i) \\
& g(i+1) = \textstyle\sum_{j=0}^{m-1} f(i-j)(1-\tau)^{j+1}
\end{aligned}
$$

If $i \leq m$, the LCS will always be less or equal to m. Thus, $f(i) = \tau, \quad g(i) = 1-\tau$. Therefore, the recursive equations for calculating $f(i)$ and $g(i)$ are

When $1 \leq i \leq m$

$$
\begin{cases}
f(i) = \tau \\
g(i) = 1-\tau
\end{cases}
\tag{3.3}
$$

When $L \geq i \geq m+1$

$$\begin{cases} f(i) = \tau(f(i-1) + g(i-1)) \\ g(i) = \sum_{j=0}^{m-1} f(i-j-1)(1-\tau)^{j+1} \end{cases} \tag{3.4}$$

The time complexity to calculate the probability for $LCS = m$ is $O(2L + mL)$. Since the sequence length is L, m can be any integers between 0 and L. To calculate the probability for each $m \in [0, L]$, the time complexity is $O(2L^2 + (L+1)\frac{L^2}{2}) = O(L^3)$. For HIV, its genome length is about 10^4. Calculating directly with equations 3.4 for all the possible LCS is time consuming. It costs hours to calculate the probabilities for all the available LCS with a typical single-core CPU.

In fact, the result of $g(i)$ can be used to calculate for $g(i+1)$, which will reduce the total time complexity to $O(L^2)$. Let $S(i) = f(i) + g(i)$, we will have the recursive relationship (Proof omitted)

$$S(i+1) = S(i) - \tau(1-\tau)^{m+1} S(i-m-1), \quad i \geq m+2 \tag{3.5}$$

Therefore, we get

$$S(i) = \begin{cases} 1, & \text{if } 1 \leq i \leq m \\ 1 - (1-\tau)^{m+1}, & \text{if } i = m+1 \\ 1 - (1+\tau)(1-\tau)^{m+1}, & \text{if } i = m+2 \\ S(i-1) - \tau(1-\tau)^{m+1} S(i-m-2), & \text{if } i \geq m+3 \end{cases} \tag{3.6}$$

It is linear to calculate the probability for one LCS. The time complextiy for calculating $LCS = m$ is $O(L)$. To calculate each $m \in [0, L]$, the time complexity is $O(L^2)$.

3.2.3 Use paired-end reads to distinguish different haplotypes

The scattered distribution of the mutations or insertions/deletions between different strains make strain-level assembly possible by employing paired-end reads information. We first quantify the strain similarity with sizes of the longest common substring (LCS). If the LCS size is smaller than read size, path finding becomes trivial because different strains correspond to different paths in the overlap graph G. The analysis of available data shows that the size of the longest common regions are larger than typical read size but smaller than fragment size, which is the end-to-end size for a read pair. Thus, it is highly likely that the two ends of a read pair can encompass the common regions between different strains. As paired-end reads are sequenced from the same fragment, they should be assembled into the same contig. Thus, by using the paired-end information, two different strains can be distinguished from each other.

Figure 3.2 sketches the basic idea behind strain-level assembly using paired-end reads. There are only two mutations between the two strains in this example. The LCS is the common region between the two mutation loci. The overlap threshold is set as half of the read size. The overlap graph G and the paired-end graph PE_G are constructed accordingly. In G, there are four long paths: $a.1 \rightarrow b \rightarrow c \rightarrow e \rightarrow f \rightarrow a.2$, $a.1 \rightarrow b \rightarrow c \rightarrow e \rightarrow f \rightarrow d.2$, $d.1 \rightarrow b \rightarrow c \rightarrow e \rightarrow f \rightarrow a.2$, and $d.1 \rightarrow b \rightarrow c \rightarrow e \rightarrow f \rightarrow a.2$. The goal of assembly is to output the two correct paths (i.e. $a.1$ to $a.2$ and $d.1$ to $d.2$). Three types of information can be used. 1): Coverage: if the two strains have highly different coverage, we may distinguish the two strains correctly. 2): Enumeration of cliques. This method was recently adopted by several tools [147, 93, 5]. For reads with sufficient coverage, reads forming cliques tend to come from the same haplotype and thus can be merged as a super-read. This process can be iteratively applied to extend local haplotype to global one. 3): paired-end information.

By using the paired-end information, a path starting with a.1 in Figure 3.2 will end with a.2 because a.1 and a.2 should be assembled into the same contig. For the same reason, a path starting with d.1 will only extend to d.2. Thus, the path finding will output two paths, correctly representing the two haplotypes. Of the three types of information, paired-end information is the most accurate one and can tolerate LCS up to the fragment size. Coverage difference has limited applications because it requires relatively uniform coverage and can only distinguish strains of highly different abundance. Clique enumeration can lead to "chimeric" super read even when the LCS is only longer than read size. In this example, the clique enumeration will not be able to distinguish the two strains. As shown in Figure 3.2.(B), there are five cliques of size 4. By merging the reads into super-reads inside each clique, we obtained five super-reads in Figure 3.2.(C). Their overlap graph is shown in Figure 3.2.(D). No matter whether cliques of size 3 are used for iterative merge, using cliques won't be able to distinguish the two strains without using paired-end information.

3.2.4 The whole pipeline of PEHaplo

There are five major components in the pipeline of PEHaplo: error correction, strand correction, overlap graph construction, path finding, and contig correction. In the first pre-process stage, reads with multiple low-quality or ambiguous base calls are filtered or trimmed. Base-calling errors or indels are corrected from the filtered set of reads using alignment-based error correction. Duplicated reads and substring reads are then removed from the corrected reads. Second, an overlap graph is built from the pre-processed reads and the strand of reads are adjusted by traversing the graph. The output reads will have the same orientation. The third stage will build the overlap graph again from the strand-adjusted reads and utilize various graph pruning methods to remove possible random overlaps and simplify the graph for efficient assembly. In the fourth stage, paired-end guided path finding algorithms are applied to produce contigs from the overlap graph. Finally, we

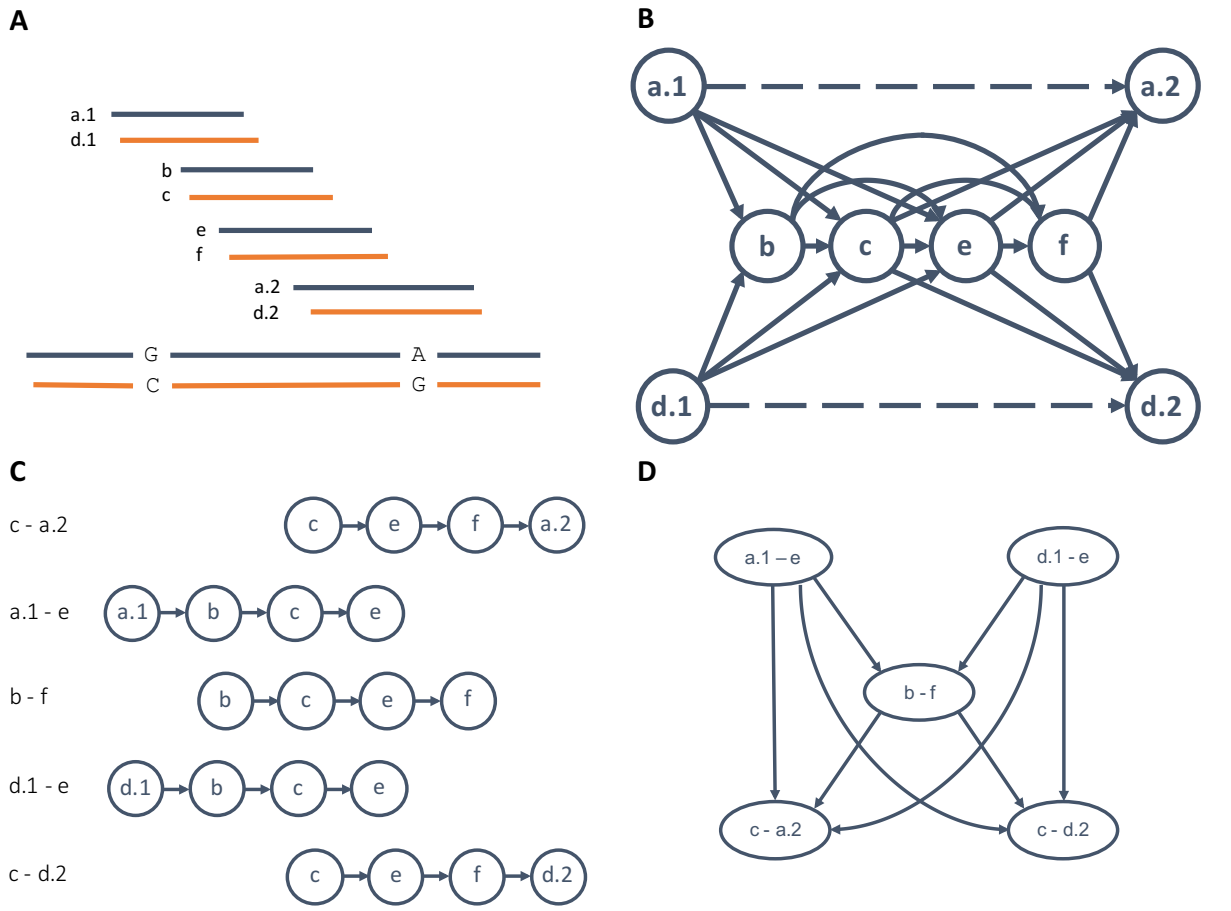

Figure 3.2: (A) The bottom two long lines (red and black) represent two haplotypes, which only differ by two mutations at two loci (G-C and A-G). Short lines represent reads sequenced from the two strains. Red reads are sequenced from red strain while black reads are sequenced from black strain. The reads are sorted by their read mapping positions against their native strain. a.1 and a.2 are a read pair from the black strain. d.1 and d.2 are the read pair from the red strain. (B) The overlap graph and the paired-end graph are combined in one figure. Nodes b, c, e, and f originate from the common region of the two strains. The dashed lines represent paired-end read connection. (C). The super-reads generated by merging reads in five cliques of size 4. Each super-read is named using the starting node and ending node. (D). The new overlap graph generated using the super-reads.

align paired-end reads against produced contigs to identify and correct potential mis-join errors.

3.2.4.1 Data pre-processing

Error correction based on coverage information, usually improves *de novo* assembly. An alignment-based error correction tool Karect [2] is used to correct substitution, insertion, and deletion errors. Besides adopting existing error correction tools, we also removed reads with very low abundance. Only reads that are duplicated at least n times in the original data set will be used for downstream analysis. Using a probability model, we show that this method can further filter out error containing reads while maintaining sufficient reads for *de novo* assembly.

Let N be the total reads number, r be the read length and L be the genome size, the probability that k reads start from the same location on the genome can be estimated by a Poisson distribution:

$$Pr[k] = \frac{\lambda^k e^{-\lambda}}{k!}, \quad \lambda = N/L \tag{3.7}$$

$$Pr[k >= n] = 1 - \sum_{i=0}^{n-1} \frac{\lambda^i e^{-\lambda}}{i!}, \quad n \geq 1 \tag{3.8}$$

We denote e_{base} be the probability of sequencing error at each base, N_e be the number of error base for one read, the probability that a read contains at least one sequencing error is $Pr[N_e \geq 1] = 1-(1-e_{base})^r$. For n reads originating from the same location on genome, the probability that they have at least one sequencing error on the same place can be calculated as $Pr = Pr[N_e \geq 1]^n / r^{n-1}$. In the HIV MiSeq data set we used (detailed description in Results), ~700k error corrected reads are left for 5 HIV strains. The genome length is ~10k bp. $\lambda = 7 \times 10^5/(5 \times 10^4) = 14$. For each base on the genome, the probability that at least 3 reads start from it is over 99.9%. If we assume $e_{base} = 0.01, \quad r = 200$, the probability that a read sequence duplicated at least 3 times but contains at least one sequencing error is $Pr \leq (1-0.99^{200})^3/200^2 \approx 1.62e-5$. The results reveal

that when sequencing depth is deep enough, keeping reads with duplications is able to filter out error containing reads and will not reduce connectivity.

3.2.4.2 Overlap graph construction

All reads remained after pre-processing are used to construct the overlap graph. A straightforward overlap detection method requires $O(n^2)$ comparisons, which is computationally expensive for large sequencing data sets. There are efficient implementation of all-pairs suffix-prefix comparison algorithms based on data structures such as hashing table or compact prefix tree [47, 51]. In PEHaplo pipeline, we first use Readjoiner [47] to compute all suffix-prefix matches among reads for read strand adjustment, and then use Apsp[51] for the final overlap graph construction.

Strand correction with overlap graph Reads in the raw data set can come from different strands, and the two reads in a read pair come from two different strands. We use Readjoiner to construct the overlap graph and traverse each node in the graph for strand adjustment. Readjoiner provides a set of fast and space efficient algorithms to compute suffix-prefix matches among all pairs of reads using suffix sorting and scanning methods. For two reads r_i and r_j, we denote r_i' and r_j' as their reverse complements. Readjoiner computes all possible overlaps between (r_i, r_j), (r_i', r_j) and (r_i, r_j'), and label the overlaps with '+ +', '- +' and '+ -' respectively if their suffix-prefix matches are longer or equal to the overlap threshold. Note that transitive edges are removed in the output by Readjoiner.

We use a breadth-first search (BFS) traversal method to adjust the strand of each read. The traversal starts at a start node (with in-degree of 0) and label it as '+', then recursively labels all its successors and predecessors based on the edge type. The '+ +' type means current node and its neighbor come from the same strand, while '- +' or '+ -' types mean a different strand. After traversing the whole graph, all the reads labeled with '-' will be replaced with their reverse

complements.

With strand-adjusted reads, we use another tool Apsp [51] to construct a new overlap graph since we will need all overlap edges this time.

3.2.4.3 Graph pruning

The original overlap graph generated from the output of Apsp is usually very complex because of the large data size, transitive edges, sequencing errors, and highly similar regions shared by haplotypes. We apply an iterative graph pruning procedure to repeatedly simplify the graph at each iteration.

Merge reads in cliques We are interested in cliques in the overlap graph because reads within a clique can share true mutations while sequencing errors are usually more random and are not shared by the majority of reads. Therefore, cliques can be used to distinguish true mutations from sequencing errors. Several recent haplotype reconstruction tools [5, 147] merge reads inside cliques as super-reads and conduct iteratively haplotype extension.

We handle cliques differently to these existing tools. Instead of directly mering each clique to a super-read, we merge a group of linked cliques simultaneously. The linked cliques is defined as a cluster of cliques which share common nodes with at least one other clique in the cluster. Each clique cluster, denoted as G_c, is a connected subgraph. We simplify this subgraph by removing transitive edges and performing collapse operations as described below and then connect G_c to the original overlap graph by reconstructing connections to its starting and ending nodes. The edges connected to other clique nodes in the overlap graph are removed. Those common nodes shared by cliques will be kept and we further apply paired-end information to identify strains in path finding step. By merging linked cliques, we are able to simplify the overlap graph while avoid resulting in "chimeric super-read" (Figure 3.2(D): $(a.1-e) \rightarrow (b-f) \rightarrow (c-d.2)$).

Remove transitive edges and node collapsing An edge $u \rightarrow v$ is called transitive if there exist other paths from u to v in the graph. Transitive edges are usually removed in graph-based assembly algorithms to simplify the graph while still keeping the connectivity. We use a depth-first search (DFS) based algorithm to remove transitive edges from the overlap graph $G = (V, E)$:

1. For each vertex $u \in G$, start DFS from each of its successor v.

2. For each vertex w that could be reached by DFS from v, remove edge $u \rightarrow w$ if the edge exists.

The overall complexity of the algorithm above is $O(|V|(|V|+|E|))$, which runs $|V|$ DFSs to remove all transitive edges.

Linearly connected nodes can be merged without loss of reachability. After transitive edge removal, the overlap graph tends to have chains of linearly connected vertices, which are collapsed to further simplify the graph.

Remove false edges using read pairs Due to the nature of virus quasispecies, different haplotypes of one species usually have very high sequence similarity (possibly over 90%), which can easily cause overlaps between reads originating from different strains. Therefore, having a suffix-prefix match does not guarantee that the two reads originate from the same virus strain. Wrong edges increase the complexity of graph and may also produce chimeric contigs. We employ paired-end information to remove potentially wrong edges.

The overlap cutoff l is an important parameter. A small l tends to keep most true overlaps but also introduces more false connected edges, while large l is likely to eliminate most false overlaps but can possibly miss true connections for reads from lowly sequenced regions. In PEHaplo, we initially choose a relatively small overlap cutoff, then apply a set of paired-end based heuristic methods to remove potential false edges. The key idea is that for an edge formed between reads from different haplotypes, these two nodes or their predecessors and successors are not usually supported by paired end connections. Thus, we will use paired-end connection as evidence to re-

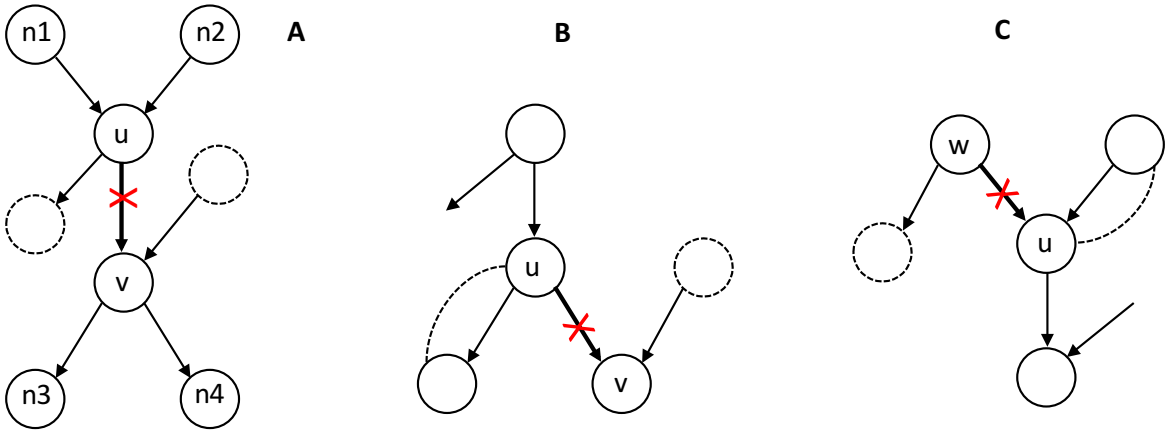

Figure 3.3: Removing false edges using paired-end information. Solid lines represent overlaps between nodes and dashed lines represent the paired-end connections between nodes. The overlap edges with red cross will be removed if insufficient paired-end information exist between their ends.

move false edges. Specifically, we focus on three cases as shown in Figure 3.3, and remove edges without paired-end information support. To aid the explanation, we introduce the following terms for the overlap graphs. Let $u \rightarrow v$ be an edge between nodes u and v. Thus, u is the predecessor of v and v is the successor of u. A node can have multiple predecessors or successors. Let function $succ(u)$ represent the set of all successors of u and let $pred(u)$ be the set of all predecessors of u. We examine edges in the following cases.

1. For an edge $u \rightarrow v$, if out-degree of u is larger than 1 and in-degree of v is larger than 1, check paired-end connections between u and v, $pred(u)$ and v, u and $succ(v)$. If no paired-end connections between these nodes and the overlap between u, v is less than a user-defined edge removal cutoff l_1 ($l_1 \geq l$), remove edge $u \rightarrow v$ (Figure 3.3(A)).

2. For a node u with in-degree being 1 and out-degree larger than 1, we check the paired-end connections between u and all nodes in $succ(u)$. For an edge $u \rightarrow v$, we remove this edge if the following three conditions are met. 1) There is no paired-end connection between u and v. 2) There exists a node $v' \in succ(u)$, ($v' \neq v$), and there is paired-end connection between u and v'. 3)

The in-degree of v is larger than 1. Intuitively, if a node u has a large number of out-going edges, it is highly possible that some of them are only random connections. Thus, we only keep the ones with paired-end supports (Figure 3.3(B)).

3. Similar to case 2, if a node has a large number of incoming edges but only has one out-going edge, we remove the ones without paired-end support. For a node u with out-degree being 1 and in-degree larger than 1, we examine the paired-end connections between all nodes in $pred(u)$ and u. For an edge $w \rightarrow u$, we remove this edge if the following conditions are met. 1) There is no paired-end connection between w and u. 2) There are paired-end connections between other predecessor nodes and u. 3) The out-degree of w is larger than 1 (Figure 3.3(C)).

3.2.4.4 Paired-end guided path finding

With sufficient coverage, virus haplotypes can be recovered by finding paths from the graph. Although the graph has been greatly simplified with previous steps, the high sequence similarity between virus haplotypes can still result in complex overlap graphs with a lot of bifurcations. To recover more accurate contigs, paired-end information of reads are widely used in many assembly tools for guiding the creation of contigs [169, 163, 93] or scaffolds. The rationale is that two ends of a read pair are sequenced from two ends of a fragment, thus should be assembled in the same contig or scaffold. Paired-end information can guide the path extension to go over common regions shared by two genomes since the length of fragment is usually much longer than the length of reads (Figure 3.4).

In our method, we use a DFS-based path extension algorithm and select the right node to append to the path with a decision tree at each bifurcation. Path finding starts at a start node (in-degree of 0) and terminates at an end node (out-degree of 0) or when meeting a visited node. At

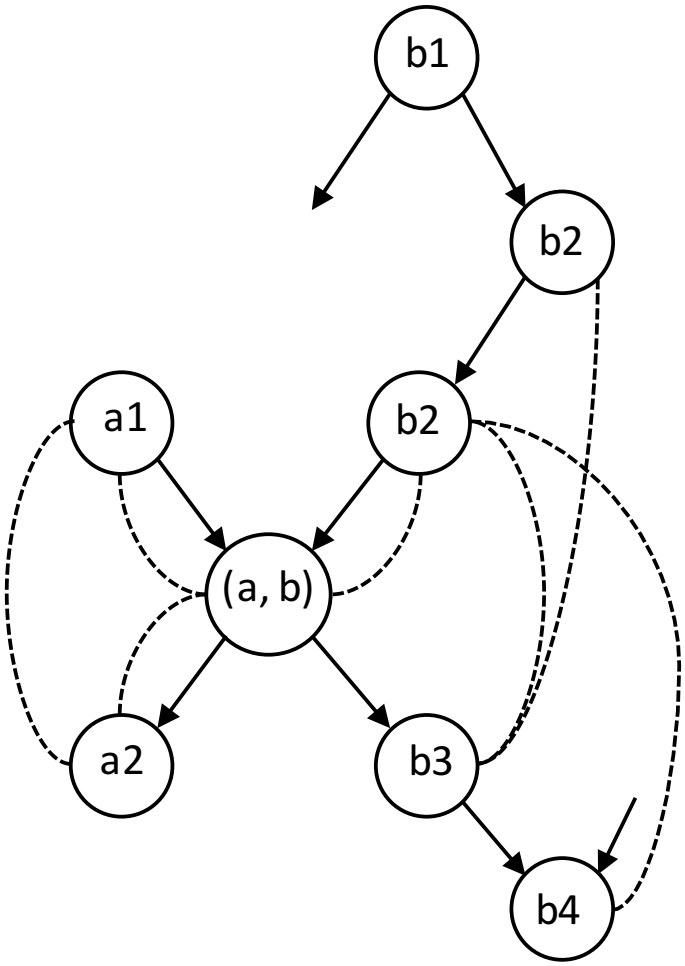

Figure 3.4: Paired-end information guide the path extension to go over bifurcation nodes. With nodes coming from two strains *a* and *b* in the figure, those nodes belonging to the same strain can be correctly combined to one path based on the paired-end connections. Solid lines represent overlaps between nodes and dashed lines represent the paired-end connections.

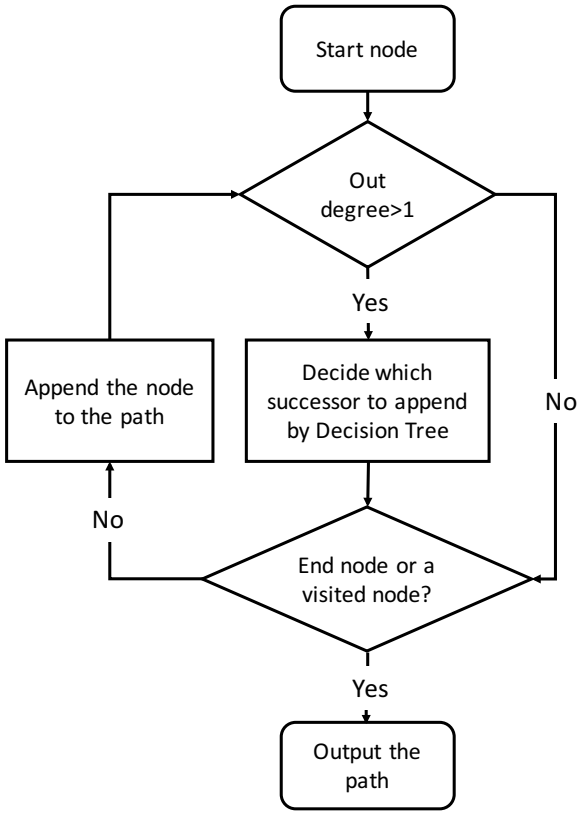

Figure 3.5: Path finding protocol. Paths are outputted when meeting an end node or a visited node.

each vertex with multiple successors, we transform the available paired-end and overlap information as features and decide which successor to append to the current path with a pre-trained decision tree. The flowchart of path finding algorithm is shown in Figure 3.5. The paired-end connections between nodes can be efficiently accessed from the constructed paired-end graph *PE_G*.

Feature design and decision tree classification To find the correct path from the overlap graph, we need to select the right node each time we extend the path. In particular, when a node has multiple successors, a choice needs to be made for path extension. In general, we choose a successor with the most number of paired-end connections with the nodes in the current path. However, different types of paired-end connections should be treated differently in distinguishing haplotypes. In particular, we need to differentiate single-in single-out nodes (SISO) from other nodes. SISO nodes have in-degree of 1 and out-degree of 1. We have high confidence that these

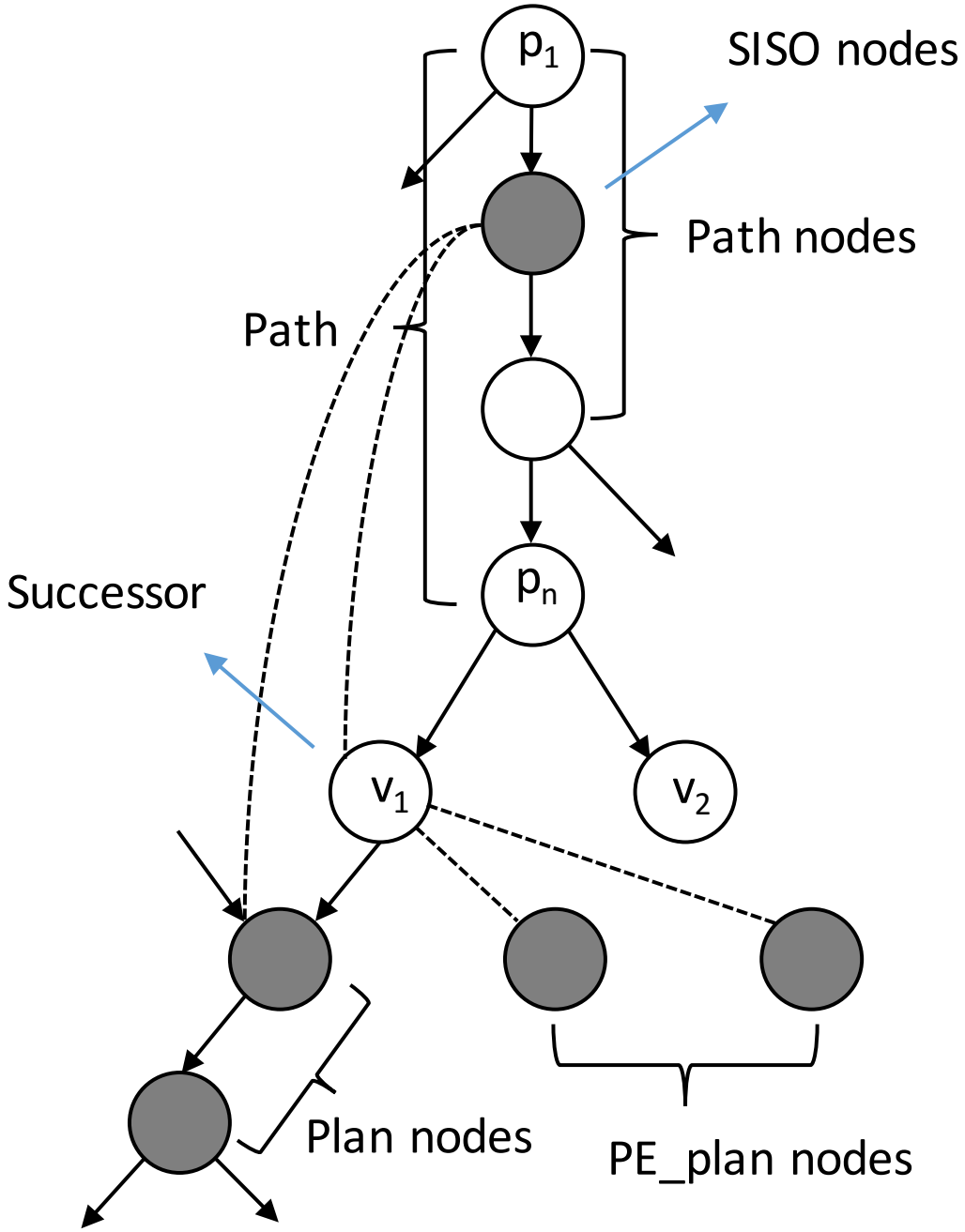

Figure 3.6: In this example as shown in the figure, the ending node p_n of current path has two successors v_1 and v_2. The solid lines are overlaps and dashed lines are paired-end connections between nodes. Five features are computed to choose the right successor to extend the path. These features are computed as paired-end graph edge weights between nodes in current path and nodes associated with the successor. As for v_1, the SISO score is calculated between SISO nodes and v_1; plan score between SISO nodes and plan nodes; PE_plan score between SISO nodes and PE_plan nodes; path score between path nodes and v_1; path_plan score between path nodes and plan nodes.

nodes belong to one haplotype and thus any paired-end connection incident to SISO nodes can recruit nodes belonging to the same haplotype . On the contrary, the nodes originating from common regions of two or more haplotypes are not SISO nodes. Paired-end connections between those nodes cannot provide useful guidance in path finding. In our feature design, we distinguish paired-end connections involving SISO nodes and other nodes.

Let PE_G be the paired-end graph. $Path = \{p_1, p_2, ..., p_n\}$ be the current path. The ending node p_n in the current path has multiple successors. For each successor v of p_n, we compute the scores of the following 5 features for node v. For each feature, we use $N_{feature}$ to recored the number of successors with score value greater than 0. The features will then be used to make a choice for path extension. To better illustrate how these features are calculated, we give an example as shown in Figure 3.6.

1. SISO score: we calculate the SISO score as the summation of edge weights in paired-end graph PE_G between SISO nodes in the current path $Path$ and v. Note that edge weight in PE_G is the number of paired-end reads between two nodes. We use N_{SISO} to denote the number of successors that have a SISO score greater than 0.

2. Plan score: similar to SISO score, we calculate the plan score as the summation of PE_G edge weights between SISO nodes and the plan nodes. The plan nodes are defined as follows. Starting from v, define v's successor as plan node if v has only one successor. Repeating this procedure until reaching a node with more than one successor or out-degree of 0. Plan score considers the paired-end read connections between all potential SISO nodes inside a path, including the children nodes of p_n. N_{plan} denotes the number of successors that have a plan score greater than 0.

3. PE_plan score: v's adjacent nodes in PE_G that are not in the path $Path$ are defined as

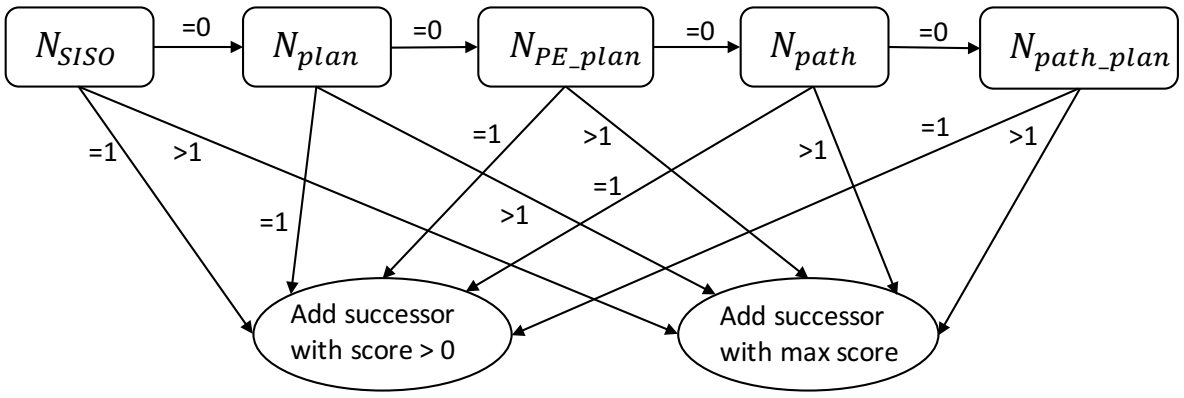

Figure 3.7: Decision tree to select the right node for extension based on paired-end score features. If $N_{feature} = 1$, we add the only node with score greater than 0 to the path. If $N_{feature} > 1$, we select the node with maximum score value. Otherwise, look at the next feature. If all the $N_{feature}$ values are 0, we will select the successor with similar reads coverage to the path.

PE_plan nodes. The PE_plan score is calculated as the summation of PE_G edge weights between SISO nodes in path $Path$ and the PE_plan nodes. N_{PE_plan} denotes the number of successors that have a PE_plan score greater than 0.

While the above three features are focused on SISO nodes, the following features extend to paired-end connections incident to other types of nodes.

4. Path score: the path nodes are all the nodes in the path $Path$ except p_n if $n > 1$ or p_1 if $n = 1$. Path score is calculated as the summation of PE_G edge weights between path nodes and v. N_{path} denotes the number of successors with path_score greater than 0.

5. Path_plan score: the summation of PE_G edge weights between path nodes and v's plan nodes. N_{path_plan} denotes the number of successors with path_plan score greater than 0.

With the 5 features shown above, we design the decision tree as shown in Figure 3.7 to classify the right node for path extension.

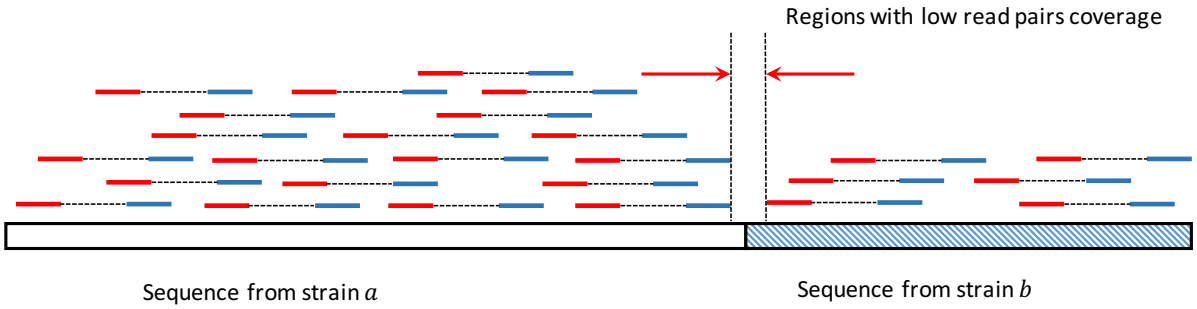

Figure 3.8: Read pair mapping profile on a misjoined contig. The contig is shown as the long bar at the bottom, which is misjoined with two sequences from strains *a* and *b*. The red and blue lines represent two ends of a read pair. Few read pairs will go across the misjoined location, thus revealing a valley in the aligned reads profile, which can be used to split the contig.

3.2.4.5 Correcting contigs with paired-end read distribution

Our methods usually generate long contigs after path finding. While paired-end guided path finding method can go over the common regions that are short than fragment length, it is limited when two strains have a LCS longer than the insert size of read pairs. To further improve the quality of assembled contigs, we apply a contig correction method similar to the tool PECC [81]. With the contigs generated after path finding, we align the raw reads to them and split contigs from the locations with low read pairs coverage (Figure 3.8).

3.3 Results

We have developed and implemented PEHaplo a *de novo* virus quasispecies assembly method based on paired-end guided path finding on overlap graph. To evaluate the performance of our method, we applied PEHaplo on both simulated HIV quasispecies data sets and HIV illumina MiSeq sequencing data set. Both of the simulated data and real data were generated from a mixture of five well-studied HIV-1 strains (HXB2, JRCSF, 89.6, NL 43 and YU2). These strains have pairwise sequence similarities from 91.8% to 97.4% (Table 3.1) and a longest common substring

of 427bp between HXB2 and NL43 (Table 3.2). We choose HIV data set because HIV-1 has high intra-patient genetic diversity. To further evaluate the efficiency of the tool, we also tested PEHaplo on a real Influenza MiSeq sequencing data.

As the haplotype sequences and compositions are known in these datasets, we are able to evaluate the performance of our methods with the assembled contigs. We also compared the produced results to recently published *de novo* assembly tools IVA [57], MLEHaplo [93] and SAVAGE [5].

Table 3.1: Pairwise sequence similarity between 5 HIV-1 strains.

	89.6	HXB2	JRCSF	NL43	YU2
89.6		93.9	91.8	93.5	93.6
HXB2			92.8	**97.4**	95.2
JRCSF				92.6	92.9
NL43					94.9
YU2					

Table 3.2: Longest common substring (LCS) between 5 HIV-1 strains. These strains have similar lengths of about 10k bp.

	89.6	HXB2	JRCSF	NL43	YU2
89.6		195	201	164	234
HXB2			180	**427**	216
JRCSF				157	201
NL43					185
YU2					

3.3.1 LCS probability simulation

To estimate the LCS between two strains within a quasispecies, we calculated the probability distribution for LCS between two strains that are n generations apart from equation 3.2 and equation 3.6. Let the mutation rate μ between two generations be 3e-5, and the genome length L be 10,000. The probability distribution for LCSs between two strains is shown in Figure 3.9. The figure shows that as the mutation rate increases, the LCS length decreases.

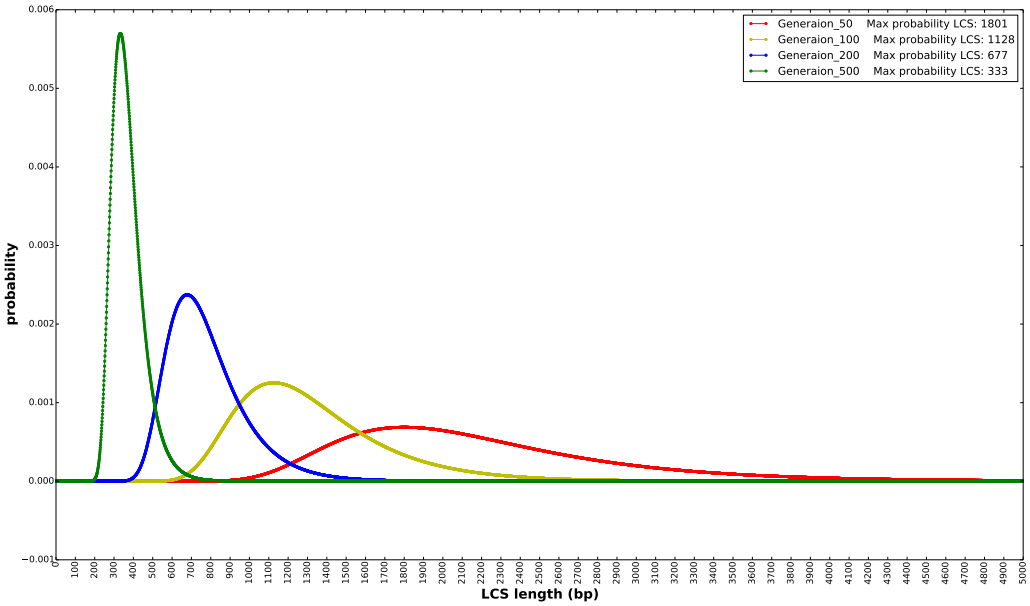

Figure 3.9: Probability distribution for LCSs between two strains that are 50, 100, 200, and 500 generations apart. The x-axis is the length of LCS, which ranges from 0 to 10,000. The y-axis is the corresponding probability for each LCS.

3.3.2 Benchmark on simulated HIV data set

We first applied PEHaplo on a simulated virus quasispecies data set. We used ART-illumina [56] to simulate 1.9e+5 paired-end, 250bp error-containing MiSeq reads from the five HIV-1 strains with average fragment length of 600bp and total coverage of 5000x. A fitness based power law equation [9] was used to simulate the coverage distribution among five strains: $C_i = bf_i^a$, where C_i and f_i denote the coverage and fitness of strain i, respectively. The coverages for each strain in the simulated data set are: 89.6 - 2190x, HXB2 - 1095x, JRCSF - 730x, NL43 - 547x, YU2 - 438x.

Following the PEHaplo pipeline, we first performed error correction and duplicated sequences removal on the raw simulated data set. With 1.9e+5 error corrected reads, 48,833 reads were kept after removing duplicates. We then only kept those reads that are duplicated at least 3 times in the raw data, further reducing the reads number to 26,961. After adjusting reads orientation,

we constructed the overlap graph with tool Apsp. The original overlap graph has 26,961 nodes and 977,570 edges, after merging linked cliques, removing transitive edges, collapsing nodes and removing bad edges, 125 nodes and 166 edges were left. We then recovered paths and generated contigs from this simplified graph based on paired-end information.

PEHaplo generated 14 contigs from the simulated data set. The generated contigs were aligned and evaluated to five reference genomes with MetaQuast [100] and the results are summarized in Table 3.3. The contigs are able to cover over 97% on the 5 virus strains, with a N50 of 7170 bp. The largest contig has a length of 9667bp, which can almost cover a whole HIV strain. Meanwhile, these contigs have low mismatch and indel rates.

We also assembled the simulated reads with benchmark tools IVA, MLEHaplo and SAVAGE and summarize their results in Table 3.3. With the default parameters, IVA produced a single, long contig from the error corrected reads. This long contig has a length of 13,434 bp and can cover the whole genome of 89.6 strain and about 43% of the HXB2 strain. The results of IVA reveal that it tends to generate one consensus genome sequence corresponding to the haplotype with the highest coverage. Other strains are largely missed, indicating that it may not fit for virus quasispecies assembly. Using k-mer size of 55, MLEHaplo produced 205 contigs that cover 78% of the five HIV-1 strains. The contigs it produced are quite fragmented, with a low N50 value of 671 bp and largest contig of 1,716 bp. Following the guidance from the tutorial, we set the overlap cutoff as 190 bp to run SAVAGE. It produces 43 contigs that cover over 99% of the reference genomes, with a N50 above 2,000 bp, and largest contigs of 9,594 bp.

Comparing to IVA and MLEHaplo, PEHaplo is able to produce longer contigs with less mismatches and indels on the simulated HIV-1 five strains data set. SAVAGE produced more contigs that could cover almost all the five strains. However, these contigs have a much lower N50 value (2,228 bp) than PEHaplo (7,170 bp).

Table 3.3: Assembly results on simulated HIV data set for IVA, MLEHaplo, SAVAGE and PEHaplo. Contigs that are at least 500 bp are aligned to the true haplotype sequences with a similarity cutoff of 98%. The N50 value is the maximal length that all contigs of at least this length cover at least half of the total assembly length.

Tools	#contigs	N50	Genomes covered (%)	Unaligned length	Mismatches (%)	Indels (%)
IVA	1	13,434	28.7	**0**	0.809	0.051
MLEHaplo	205	671	78.0	81,125	0.542	0.008
SAVAGE	43	2228	**99.4**	5285	0.019	0.002
PEHaplo	14	7170	97.8	**0**	**0.015**	**0**

3.3.2.1 Paired-end guided path finding is able to generate accurate long contigs

In our algorithms, we have applied multiple methods to simplify the overlap graph and collapse nodes before path finding. While these steps greatly reduce the complexity of the graph, they cannot distinguish different strains that share long common regions. Thus, the path finding algorithm based on paired-end connections play a crucial role for producing high quality long contigs from the overlap graph.

To evaluate the performance of our path finding algorithm, we assembled the sequences in the overlap graph before path finding with the popular assembly tools IDBA-UD [115] and Ray Meta [14]. We then aligned the generated contigs to reference genomes and compared with the results produced by PEHaplo. The results reveal that those contigs assembled by IDBA-UD and Ray Meta from the reduced overlap graph are fragmented and may contain many misjoined segments: they cover large proportions of the five reference genomes, but with high mismatch and indel rates and their average lengths are much shorter than PEHaplo. The experiment shows that the paired-end guided path finding algorithm in PEHaplo is able to correctly recover long haplotype sequences from the virus quasispecies sequencing data.

3.3.3 Benchmark on MiSeq data set

To further assess the performance of assembly methods, we applied PEHaplo on a real HIV quasispecies data set (SRR961514) sequenced from the mix of the same five HIV-1 strains as described above with Illumin MiSeq sequencing technology [39]. This data set contains 714,994 pairs (2x250 bp) of reads that cover the five strains to 20,000x.

We used PEHaplo to perform similar processing procedures on the real HIV quasispecies data. With 774,044 filtered and error corrected reads, 98,947 reads were kept after removing duplicates and substrings. Since the raw data set has extremely high coverage on the five strains, we still kept those reads that are duplicated at least three times in the raw data set. After these pre-processing procedures, 26,691 reads were kept for strand adjustment and assembly.

PEHaplo produced 33 contigs from the real MiSeq HIV data set that can cover over 92% of the five HIV-1 strains. These contigs have a N50 value about 2,500 bp and the longest contig is 9108 bp. The results are summarized in Table 3.4. Compared to simulated HIV data set, PEHaplo has generated more contigs but with a lower N50 value and higher mismatches and indels on the real data set. We notice that the real HIV data set contains more sequencing errors and has a more variable insert size than the simulated data set.

We again compared the performance of PEHaplo with IVA, MLEHaplo and SAVAGE. IVA generated 10 contigs that cover about 20% of the five strains. These contigs still cover longer parts on haplotypes with higher sequencing coverage. But they spread to four strains this time, likely because the five strains have close sequencing coverages in the real data set. With the same parameters as above, MLEHaplo produced 234 contigs that can cover over 53% of the five genomes with similar mismatch and indel rates to the simulated data set. Strikingly, it generated much longer contigs on the real data, with a N50 value of 6,501 bp and the largest contig of 8,470

bp. However, these contigs contain many misjoined segments. Over 150 contigs with total length of 787,272 cannot align to any reference genomes. Since the SAVAGE paper [5] has shown their results on the same data set, we use the metrics in their literature for evaluation. From their results, SAVAGE produced 482 contigs that cover over 90% of the reference genomes, with a N50 of 1,062 bp, and largest contig of 4,256 bp (Table 3.4).

On the real HIV data set, PEHaplo can still produce longer contigs with fewer mismatches and indels than all three benchmarked tools. Overall, PEHaplo is able to assemble a bunch of reads that are sequenced from multiple virus strains sharing high similarities, generate long, high quality sequences and recover most of the target haplotypes. The tool consumes less running time while still produces high quality contigs. Compared to other state-of-the-art methods, our tool usually produces fewer but longer contigs. In figure 3.10, we show the contigs alignment result on HXB2 strain for PEHaplo and SAVAGE. Those contigs were aligned to HXB2 strain with a similarity cutoff of 98% by MetaQuast evaluation tool. This figure shows that while both tools produced contigs that cover the virus genome to a similar proportion, PEHaplo was able to generate less but longer contigs.

Table 3.4: Assembly results on real HIV MiSeq data set for IVA, MLEHaplo, SAVAGE and PEHaplo. Contigs are evaluated with MetaQuast using the same parameters to simulated data set.

Tools	#contigs	N50	Genomes covered (%)	Unaligned length	Mismatches (%)	Indels (%)
IVA	10	1150	20.1	1150	0.660	0.052
MLEHaplo	234	6501	53.6	786,272	0.588	**0.035**
SAVAGE	482	1062	90.5	**0**	0.147	0.048
PEHaplo	33	2553	**92.4**	**0**	**0.117**	0.049

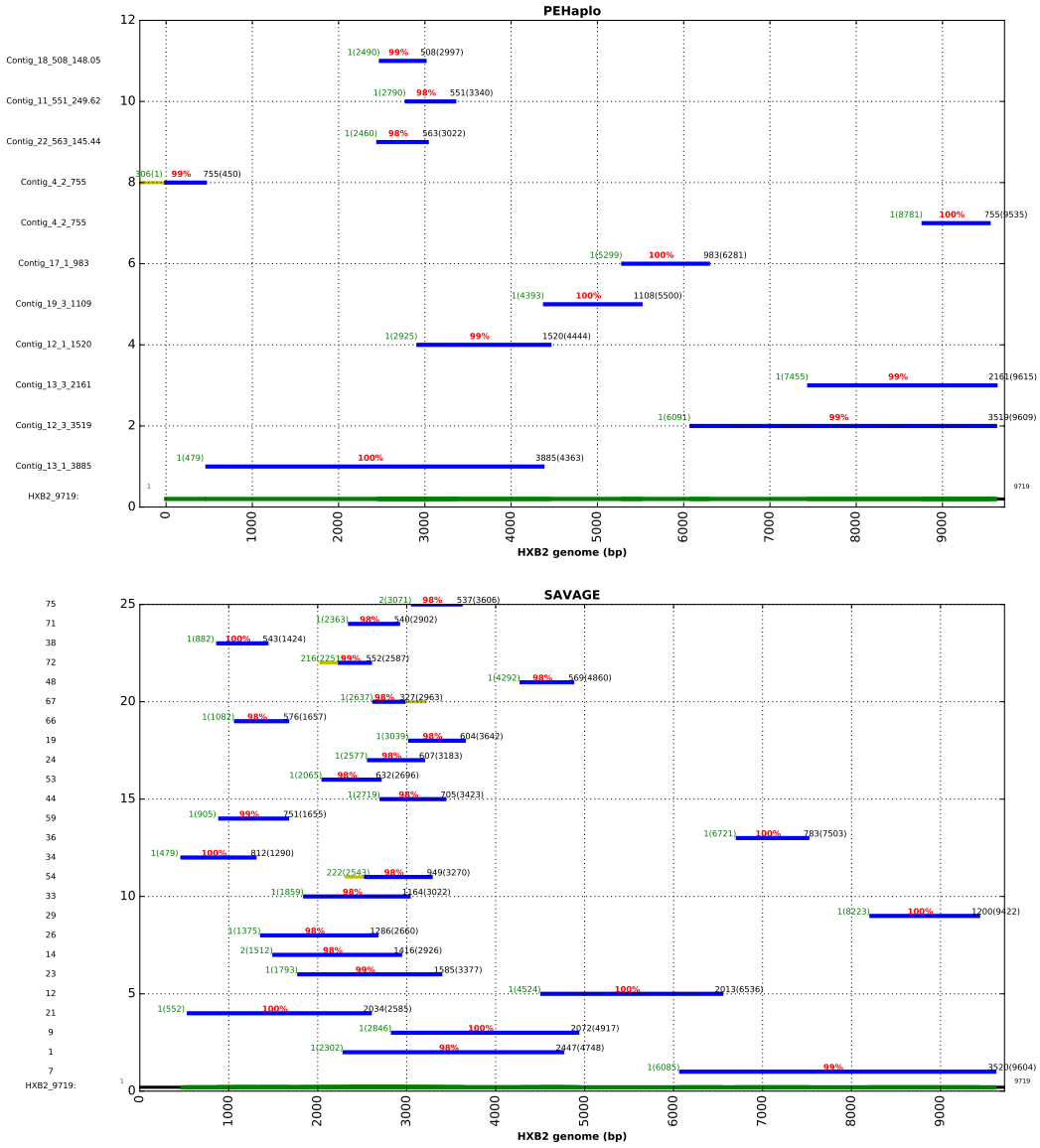

Figure 3.10: Contigs alignment result on HXB2 strain for PEHaplo and SAVAGE. These contigs were produced from the real HIV MiSeq data and aligned to reference genome with MetaQuast. The x-axis is the coordinations of HXB2 genome, with the regions covered by contigs in green and others in black. The y-axis represents the number of contigs, with contig names listed on the left panel. On each contig, the green number at the left is the starting coordinate of the aligned contig and the number inside of the parenthesis shows the starting coordinate on the reference genome, the black value at the right is the ending coordinate of the aligned contig and the number inside of the parenthesis shows the ending coordinate on the reference.

3.3.4 Bechmark on simulated biased HIV data sets

To evaluate the performance of our methods on assembling low abundance strains, we used HIV strains HXB2 and NL43 to simulate three groups of data sets with extremly biased coverages between them. The total coverage for each group 1000x, with HXB2-900x, NL43-100x; HXB2-950x, NL43-50x; and HXB2-990x, NL43-10x for each group, respectively. These data sets were also simulated by ART-illumina with MiSeq platform.

With the similar processing procedures on HIV 5 strains data, we used PEHaplo to assemble contigs from these data sets and compared the results with SAVAGE, which works much better than MLEHaplo and IVA. The results are shown in Table 3.5.

Table 3.5: Assembly results on simulated biased HXB2-NL43 MiSeq data set fo SAVAGE PE-Haplo. Contigs are evaluated with MetaQuast.

HXB2-NL43 coverages	Tools	#contigs	N50	Genomes covered (%)	Unaligned length	Mismatches (%)	Indels (%)
900:100	SAVAGE	7	2500	46.76	581	0.022	0
	PEHaplo	11	8163	80.26	0	0.038	0
950:50	SAVAGE	8	8032	46.76	1817	0	0
	PEHaplo	1	9470	46.76	0	0.033	0.011
990:10	SAVAGE	13	2130	46.75	1590	0.022	0
	PEHaplo	1	9509	48.95	0	0	0

The results reveal that both tools failed to assemble the rare strain with 5% or 1% abundance. However, PEHaplo was able to better assemble the dominant strain with producing one long contig. In addition, when the rare strain constituted 10% (100x) of the total coverage, PEHaplo could partially assemble it, while SAVAGE only assembled the dominant one.

3.3.5 Bechmark on Influenza data set

In addition of HIV data, we also applied PEHaplo on a real Influenza H1N1 data set (SRR1766219) sequenced from the mix of a wild type (99%) and a mutant type (1%). This data set is sequenced with Illumina MiSeq sequencing technology, containing 646,879 pairs (2x250) of reads that cover the two strains to 23,000x. The mutant type carries two silent mutations in the M1 ORF (C354T and A645T, segment 7).

We first perform similar pre-processing on the Influenza data. With 851,988 filtered and error corrected reads, 27,888 reads were kept after removing duplicates and substrings. We still kept those reads that duplicate at least three times in the raw data set. After pre-processing, 11,940 reads were kept for strand adjustment and assembly.

PEHaplo produced 10 contigs from the MiSeq Influenza data, with 8 contigs cover over 99% of the 8 segments of Influenza genome and 2 contigs unaligned. Again we compared the assembled results with SAVAGE. The results are summarized in Table 3.6. On Influenza data, SAVAGE has produced 220 contigs with a N50 value of 620 bp. These contigs are able to cover over 96% of the Influenza genome. The results show that PEHaplo works much better than SAVAGE on Influenza data as it successfully assembled all the 8 segments.

Table 3.6: Assembly results on Influenza MiSeq data set for SAVAGE and PEHaplo. Contigs are evaluated with MetaQuast using the same parameters to HIV data sets.

Tools	#contigs	N50	Genomes covered (%)	Unaligned length	Mismatches (%)	Indels (%)
SAVAGE	220	620	96.3	38303	0.818	0.046
PEHaplo	10	1790	99.5	1270	0.836	0.007

3.3.6 Computational time and memory usage

To evaluate the computational cost of our tool, we also compare the running time and memory usage with the three benchmark tools. PEHaplo used 43 minutes with a peak memory usage of 2.9 GB on simulated data and 19 minutes with a peak memory usage of 1.3 GB on real data set. IVA runs very fast: it took about 17 minutes on both simulated and real data set, with peak memory usage about 2.6 GB. MLEHaplo and SAVAGE are much slower. MLEHaplo consumed over 10 hours with peak memory usage of 6.4 GB on simulated data set and about 4 hours with peak memory usage of 2.4 GB on real data set. SAVAGE used 54 minutes for simulated data and 1331 minutes for real data. It is able to keep a low memory usage of 0.5 GB as shown in their paper [5].

3.4 Discussion and Conclusion

Long reads will reduce the complexity of *de novo* assembly of general metagenomic data, including virus quasispecies data. For paired-end reads, one may consider to combine short reads into a relatively longer read before conducting assembly. We actually applied existing read joining tools for this purpose. However, joining reads is not a trivial problem as the overlapping part of the read pairs may not always be identical. Thus, existing methods of joining two ends may introduce errors. In addition, merging paired-end reads will discard the paired-end information for guiding the path finding process. As a result, the experimental results using PEAR [167] and other end merging tools show inferior performance. Thus we did not include that step in our pipeline.

The third-generation sequencing platforms such as PacBio can produce very long reads, which can cover the whole length of viral genomes. However, the high sequencing error rate (about 10%) and the lower throughput than Illumina still hamper their wide application for metagenomic sequencing. The advantages and limitations of applying current long reads technologies for

virus haplotypes reconstruction are discussed in BAsE-Seq [53]. With the increased read quality, long read sequencing technologies will greatly simplify the assembly methods for metagenomic data [39]. However, at this moment, virus haplotype reconstruction using short reads is still needed.

If the distribution of fragment size is known, we can further improve the accuracy of path finding and false edge removal. For example, with known fragment size, we can accurately compute how far we should examine the successors or predecessors for false edge removal. Currently, for computational efficiency, we did not incorporate fragment size distribution in these two steps.

Our method can be extended to metagenomic data if the member species' genomes have common regions with length smaller than fragment size. However, our analysis has shown that many genes in metagenomic data can have LCS sizes much greater than typical fragment size. For those metagenomic data, large insert sizes should be chosen for the sequencing protocol.

In conclusion, we presented a *de novo* virus haplotype reconstruction tool for viral quasispecies. We applied it to both simulated and real quasispecies data and achieved better results than several benchmarked tools.

Chapter 4

Alignment windows based viral strain-level contigs binning

4.1 Introduction

After contigs of different viral strains being assembled from viral quasispecies data, it is still unknown that how many viral haplotypes are there in the quasispecies and what are there corresponding abundances. Therefore, another crucial step in recovering viral haplotypes from metagenomic data is the estimation of number of viral haplotypes and classification of contigs assembled into different groups, which is often referred to as binning. The general binning for microbial samples is defined as further investigating the taxonomic structure of contigs and clustering/classifying them into operational taxonomic groups. For viral quasispecies assembly, these groups represent composite strains of an individual viral species that comprise a viral quasispecies.

Many binning methods exist to bin assembled contigs from metagenomic data [160, 88]. These methods usually estimate the bin number by aligning metagenomic data to a pre-established marker gene database, and then assign assembled contigs to different bins using sequence composition information and read coverage levels. For example, MaxBin [160] uses both tetranucleotide frequencies and contig coverage levels to assign assembled contigs into different bins.

In addition, there are also methods of binning contigs using the coverage profiles of the contigs across multiple metagenomic samples. The idea is that if two contigs are from the same bin, their

coverage profiles across multiple samples should be highly correlated. For example, COCACOLA [88] incorporates sequence composition and read coverage across multiple samples for binning.

While these binning tools exist, binning of contigs in a viral quasispecies has its unique challenges, and those existing tools are either not applicable or do not perform well. (1) Viral haplotypes in a quasispecies belong to the same species, and they share the same marker gene. Aligning the reads to marker gene database will only identify one species but not the number of haplotypes. (2) Viral haplotypes in a quasispceis usually share high sequence similarity (may over 90%). The sequence compositions for different haplotypes are thus very similar, and can hardly be used to differentiate them. 3) There may not be multiple samples for a viral quasispecies.

Here we present VirBin, a method designed specifically for binning contigs assembled from a viral quasispecies data. It takes advantage of contigs alignment and relative contig abundances in aligned windows to accurately estimate the number of haplotypes and cluster contigs into different groups. This method works with a single sequencing sample and does not require reference genomes. VirBin was applied on two simulated datasets and a real dataset ,and benchmarked with the recent approach MaxBin. The results show that VirBin reveals superiority in terms of both precision and recall.

4.2 Methods

4.2.1 Problem definition

A viral quasispecies is composed of a set of highly similar viral strains with different abundance levels. With assembled contigs from a quasispceis, the objective is to (1) estimate the number of haplotypes in the quasispecies; (2) cluster contigs into groups so that contigs from the same viral strain will be grouped together; (3) calculate the abundance for each viral strain in quasispecies.

Mathematically, the problem can be defined as: Given contigs $C_0, C_1, ..., C_n$ from a viral quasispecies, the goal is to estimate the number of haplotypes N, cluster the contigs into $N+1$ groups (one more group as undefined) so that contigs from the same strain will be in the same group, and calculate the abundance for each haplotype.

Since viral strains in a quasispecies belong to the same species are highly similar to each other, aligning them to marker genes usually only identifies the species. Therefore, canonical binning methods based on marker genes do not apply for viral quaspecies. However, taking advantage of the high sequence similarities between haplotypes, contigs from different strains but derived from the same genomic locations can be aligned together. With the alignment of contigs, VirBin uses a window-based method to estimate the haplotype number and more accurately calculate the relative abundance for each contig. The window is defined as a continuous region in the alignment with the same number of contigs. This method begins with windows identification from contigs alignment. Then high-quality windows, which have a high probability of containing contigs from all haplotypes, are identified. The consensus heights (number of contigs h) in these windows are used to estimate the number of haplotypes N. For each window, the relative abundance levels for contigs can be calculated from the reads alignment profiles.

4.2.2 The VirBin algorithm overview

The overall pipeline of our method is shown in Figure 4.1. There are mainly two steps: (1) align contigs and identify windows; (2) calculate relative abundances in each window and apply an expectation-maximization method to cluster the contigs.

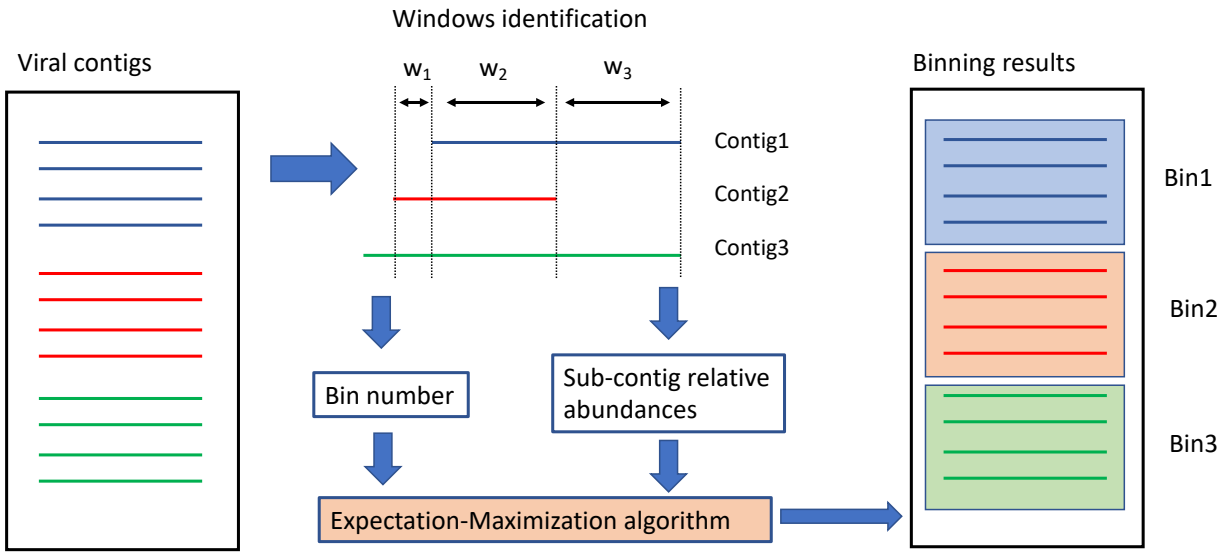

Figure 4.1: The workflow of VirBin.

4.2.3 Estimate haplotype number by contigs alignment and windows extraction

In the first step, contigs are aligned with each other using blast [20]. Each time a contig C_i is chosen as a reference contig, and all other contigs are aligned against the reference to generate an alignment profile similar to multiple sequence alignment. Although blast performs local alignment, the raw alignment results were filtered where two types of alignment relationships are allowed between every two contigs. One is overlap, where one contig's suffix align with the other one's prefix (Figure 4.2(A)). Another one is inclusion, where one whole contig aligns to a substring of another contig (Figure 4.2(B)). In this way, random alignments or alignment between repeated sequences can be avoided. Then starting from 5' end of C_i, windows are identified as the continuous regions with the same number of contigs. The algorithm for identifying windows from contigs alignment is shown in Algorithm 3.

Ideally, if the contigs are complete and all alignments between them are correct, the height of

Figure 4.2: Two kinds of alignment are allowed between two contigs to . (A) Overlap. (B) Inclusion

Algorithm 3 Windows identification from contigs alignment profile

Function: Aligned contigs depth $Height(T)$
Input: Contigs alignment result on a reference contig C, with index from 1 to L;
Output: Windows on contig C, a list of tuples

```
1: i ← 1
2: j ← 2
3: start ← 1
4: end ← 0
5: while i ≤ L do
6:     if i == L or Height[i] != Height[j] then
7:         end ← i
8:         Windows += [start, end]
9:         start ← j
10:        i++
11:        j++
12:    end if
13: end while
```

each window (h) can be used to represent the haplotype number in the quasispecies. In practice, the contigs may not cover every region of haplotypes and there may exist chimeric contigs. Therefore, we use the consensus window height as the number of haplotypes.

4.2.4 Expectation-Maximization method for binning contigs

The raw reads can be aligned to contigs to calculate their abundance levels. However, due to the potential incompleteness of the assembled contigs and high sequence similarity between viral haplotypes, directly calculating contig abundances is not accurate. Because reads from strains with missing contigs may be mapped to contigs from other strains, which will eventually lead to inaccurate estimation of contig abundances. With identified windows, denote the region of a contig inside of a window as "sub-contig" c. The position-specific aligned reads profile on each sub-contig can be used to calculate the "relative" abundances for each of them within a window, which is a more accurate estimation for the contig abundances. The average relative abundance (denote as $\bar{c}$) for a sub-contig c_i in a window is calculated as

$$\bar{c_i} = \frac{S(c_i)}{\sum_{j=1}^{h} S(c_j)} \tag{4.1}$$

where $S(c_i)$ is the summation of reads coverage on sub-contig c_i. The rationale of this method is that while the abundance estimation still may not be accurate in windows with missing contigs, we can have more accurate abundance estimation for windows with correct number of contigs. Using the relative abundances in these "correct" windows for clustering, we can have more accurate binning results.

Similarly, we can calculate the position-specific relative abundance (denote as $\vec{c}$) for a sub-contig c_i as

$$\vec{c_i}[k] = \frac{c_i[k]}{\sum_{j=1}^{h} c_j[k]}, k = 1..w \tag{4.2}$$

where $c_i[k]$ represents the reads coverage at the k position of sub-contig c_i. w is the window width.

Let the haplotype number estimated by windows height as N. With sub-contigs c, the position-specific reads coverage profile $\vec{c}$, and the average abundance $\bar{c}$, VirBin utilizes the position-specific relative abundances of sub-contigs in windows with height to estimate the probability that a contig belongs to a bin with an Expectation-Maximization (EM) method. Let the haplotypes be $H_1, H_2, \ldots, H_N$, their abundances $x_1, x_2, \ldots, x_N$ be random variables and follow distributions $E_1(x), E_2(x), \ldots, E_N(x)$, respectively. Let C be the originating contig of c_i, in total n sub-contigs from C. We used EM algorithm to maximize the posterior probability $P(c_i \in H_j|\bar{c_i})$. The algorithm contains four steps as shown below:

1. Initialize N groups by randomly assign sub-contigs to them or with guidance. The sub-contigs in the same window are assigned to different groups.

2. **Expectation**

For each group, the compoent sub-contigs' relative abundance profiles $\vec{c}$s are aggregated to calculate the empirical probability density function $E(x)$. The aggregation is performed by calculating the normalized histograms for these relative abundance profiles, so that the summation of histogram values will be 1. The likelihood that c_i being produced from a haplotype can be denoted as $P(x_j = \bar{c_i}|c_i \in H_j)$. It can be caclulated as $E_j(\bar{c_i})$.

The prior probability $P(c_i \in H_j)$ is same as $P(C \in H_j)$, which is calculated as $\sum_{k=1}^{n} P(c_i \in H_j|\bar{c_i})P(c_i)$. Here, $P(c_i)$ is the prior probability of c_i, which is calculated as $\frac{length(c_i)}{length(C)}$, $P(c_i \in H_j|\bar{c_i})$ is the posterior probability that c_i belongs to haplotype H_j from last iteration.

With both likelihood and prior, the expected probability that c_i belongs to haplotype H_j can be

calculated as $likelihood * prior$, that is

$$P^{t+1}(c_i \in H_j|\bar{c_i}) = P^t(x_j = \bar{c_i}|c_i \in H_j) * \sum_{k=1}^{n} P^t(c_i \in H_j|\bar{c_i}) \frac{length(c_i)}{length(C)} \tag{4.3}$$

3. **Maximization**

With the posterior probabilities calculated for each group distribution, we can reassign the sub-contig c_i to the haplotype with the maximum posterior probability or using Gibbs sampling. The same reassigning procedures are applied for all the sub-contigs. With the assignment results, the distribution $E(H_j)$ and prior probability $P(c_i \in H_j)$ can be updated.

4. Iterate step 2 and 3 until the clustering results do not change or the maximum number of runs have been achieved. The default maximum number of runs is 100.

4.3 Results

4.3.1 HIV simulated data set

VirBin was firstly tested on two simulated HIV quasispecies datasets for its effectiveness. One simulated data set have 5 HIV haplotypes and the other have 10.

4.3.1.1 Data simulation

HIV haplotypes simulation Multiple HIV-1 strains are available on HIV sequence database (https://www.hiv.lanl. However, the sequence similarities between them are usually below 90%, which are lower than sequence similarities between viral strains in a quasispceis. Therefore, we downloaded four HIV strains (FJ061, FJ064, FJ065, and FJ066) from the database, and generate simulated haplotypes from them by randomly mutating bases at randomly selected locations. For the viral quasispecies

with 5 haplotypes, we generated 3 simulated haplotypes from FJ061 strain and mixed them with FJ066 to compose a quasispecies (denote as 5 HIV haplotypes). The sequence similarities between each simulated haplotype and FJ061 is 97%. For the quasispecies with 10 haplotypes, 2 simulated haplotypes were generated from each FJ061, FJ065, and FJ066 strains. They were mixed with FJ064 to compose the quasispecies (denote as 10 HIV haplotypes).

Contigs simulation While viral contigs can be assembled from simulated reads with available assembly tools, the quality of generated contigs depends on the assembly algorithms and parameters used. To focus on the binning problem, we simulated error-free contigs directly from the reference genomes. For each reference genome (denote its length as L), we randomly generated a list of 20 location pairs (p_1, p_2), where $1 \leq p_1 < p_2 \leq L$ and $p_2 - p_1 + 1 \geq 500$. Each location pair represents a candidate contig. These pairs were then sorted by p_1 and unqualified pairs were filtered out from the list. The filtration starts from looking at the second pair in the sorted list (the first pair is always kept), if it has an overlap less 100 bp with the previous pair and is not a substring of the previous pair, keep it and move to the next pair. Otherwise, remove this pair from the list and move to the next. After filtration, the contigs were then generated from the reference genome with left location pairs.

HIV reads simulation With available HIV haplotypes, simulated reads were generated from them by ART-illumina [56] as error-containing MiSeq paired-end reads, with read length of 250bp, average insert size of 600 bp, and standard deviation of 150 bp. The abundance of each haplotype is calculated based on a power law equation [9]. The total sequencing depths for the 5 HIV haplotypes are 1000-x, with 438-x, 219-x, 146-x, 109-x, and 88-x for each haplotype, respectively. The sequencing depths for 10 HIV haplotypes are 2000-x, with 683-x, 341-x, 228-x, 171-x 137-x, 114-x, 98-x, 85-x, 76-x, 68-x for each haplotype, respectively. In total, there are 38,914 simulated reads for 5 HIV haplotypes, and 76,974 reads for 10 HIV haplotypes.

Table 4.1: Haplotype abundances for simulated 5 HIV haplotypes calculated from reads mapping.

Haplotype	FJ061	FJ061-h1	FJ061-h2	FJ061-h3	FJ066
Abundance(%)	43.90	21.95	14.63	10.93	8.59

4.3.1.2 Results for 5 HIV haplotypes

Haplotype abundances

By mapping simulated reads to the simulated 5 HIV haplotypes by bowtie2, we were able to estimate the abundance for each haplotype as shown in Table 4.1.

Windows identification

With the available simulated contigs and reads, we first aligned contigs to each other by blast and identify windows by VirBin. For 5 HIV haplotypes, VirBin generated 121 windows from all the contigs. The windows were sorted by their lengths and the top 5 windows are shown in Figure 4.3.

The results reveal that the height of these high-quality windows can be used to accurately estimate the number of haplotypes in the quasispecies. For example, out of the top 100 windows, 50 contain 5 contigs, 8 contain 6 contigs, and 32 contain 4 contigs. Left windows contain 2 or 3 contigs. Out of the top 50 windows, 39 have 5 contigs, 1 contains 6 contigs, and 8 contain 4 contigs. Out of the top 25 windows, 20 contain 5 contigs, 1 contain 6 contigs, and 3 contain 4 contigs. Applying a simple voting process on the top-k windows, the number of haplotypes can be estimated as the height of most abundance windows, which is 5.

By aligning reads to these contigs with Bowtie2, the relative abundance levels for each contig within a window can be calculated following the equation 4.1. Relative abundances for contigs in the top 5 windows are shown in Figure 4.3.

EM clustering

With the relative abundances calculated for contigs within windows, we applied the EM clustering

```
>Window reference_contig: FJ061_2_9136_9736     5.0     493     3599    FJ066   0.0865095479195
FJ061_1_169_9138        0.4369  1448118.0       FJ061   0.392531290961
FJ061-h1_1_327_5015     0.2215  734426.0        FJ061-h1        0.233236526254
FJ061-h2_1_467_5731     0.1524  505697.0        FJ061-h2        0.159603176643
FJ061-h3_1_516_3768     0.1046  347899.0        FJ061-h3        0.128119458223
FJ066_1_658_4564        0.0845  281238.0        FJ066   0.0865095479195
>Window reference_contig: FJ061_1_169_9138      5.0     338     3441    FJ066   0.0865095479195
FJ061_1_169_9138        0.4368  1446545.0       FJ061   0.392531290961
FJ061-h1_1_327_5015     0.2215  733816.0        FJ061-h1        0.233236526254
FJ061-h2_1_467_5731     0.1524  505304.0        FJ061-h2        0.159603176643
FJ061-h3_1_516_3768     0.1046  347628.0        FJ061-h3        0.128119458223
FJ066_1_658_4564        0.0846  281182.0        FJ066   0.0865095479195
>Window reference_contig: FJ061-h1_1_327_5015   5.0     199     3301    FJ066   0.0865095479195
FJ061_1_169_9138        0.4367  1446021.0       FJ061   0.392531290961
FJ061-h1_1_327_5015     0.2215  733614.0        FJ061-h1        0.233236526254
FJ061-h2_1_467_5731     0.1524  505173.0        FJ061-h2        0.159603176643
FJ061-h3_1_516_3768     0.1046  347537.0        FJ061-h3        0.128119458223
FJ066_1_658_4564        0.0846  281162.0        FJ066   0.0865095479195
>Window reference_contig: FJ061_2_9136_9736     5.0     6563    7482    FJ066   0.0865095479195
FJ061_1_169_9138        0.4805  457574.0        FJ061   0.392531290961
FJ061-h1_2_5011_7651    0.1928  184977.0        FJ061-h1        0.233236526254
FJ061-h2_3_6692_9618    0.1342  128050.0        FJ061-h2        0.159603176643
FJ061-h3_3_6066_8674    0.1126  107630.0        FJ061-h3        0.128119458223
FJ066_2_4602_8138       0.0793  75699.0 FJ066   0.0865095479195
>Window reference_contig: FJ061_2_9136_9736     5.0     4847    5562    FJ066   0.0865095479195
FJ061_1_169_9138        0.4801  375209.0        FJ061   0.392531290961
FJ061-h1_2_5011_7651    0.1923  151695.0        FJ061-h1        0.233236526254
FJ061-h2_1_467_5731     0.1365  106786.0        FJ061-h2        0.159603176643
FJ061-h3_2_3976_5962    0.1029  80423.0 FJ061-h3        0.128119458223
FJ066_2_4602_8138       0.0871  68016.0 FJ066   0.0865095479195
```

Figure 4.3: Top 5 windows output from simulated contigs alignment for 5 HIV haplotypes. Each line starting with '>' represents one window. The following columns represent reference contig name, window height, window starting position on reference contig, window ending position one reference contig, reference haplotype, and haplotype abundance, respectively. The rows after '>' line show the information for each contig inside of this window. The columns represent contig name, relative abundance, summation of reads coverage, reference haplotype, and haplotype abundance, respectively.

Table 4.2: EM clustering results on simulated 5 HIV haplotype contigs for VirBin and MaxBin.

	virbin		MaxBin	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
FJ066	84.6	92.9	31.4	52.5
FJ061	96.6	89.0	0.0	0.0
FJ061-h1	88.2	82.9	0.0	0.0
FJ061-h2	91.8	99.6	0.0	0.0
FJ061-h3	92.7	89.0	0.0	0.0

method in VirBin to cluster contigs into 5 groups. Since the ground truth for which haplotype each contig belongs to is know, we were able to evaluate the clustering results by calculating the precision and recall a the base level. The evaluation results are shown in Table 4.2.

The results were benchmarked with MaxBin, which is a binning tool for metagenomic scaffolds/contigs based on tetranucleotide frequencies and reads coverage levels. The MaxBin program requires marker genes to identify seed contigs for binning, which is not applicable for viral quasispecies since the different viral haplotypes belong to the same species and have the same marker gene. However, we were still able to run the core clustering program of MaxBin by providing the seed contigs manually. We randomly chose one contig from each haplotype as the seed contigs and calculated the contig abundances by mapping reads to them. The results from MaxBin are also shown in Table 4.2. For all the 24 contigs, it assigned 17 to 5 clusters, leaving 7 unassigned. One contig was correctly assigned to the cluster corresponding to FJ066 strain, but other contigs were not correctly clustered.

StrainPhlAn [152] is also a tool to characterize the genetic structure of viral strains in metagenomes. It takes the raw sequencing reads and MetaPhlAn2 [151] database of species-specific reference sequences as input and aims to output the most abundant strain for each sample. However, it failed to detect any viral species at the first step running MetaPhlAn2. ConStrains [89] is another tool designed to identify conspecific strain structures from metagenomic sequence data. It uses bowtie2

Table 4.3: Haplotype abundances for simulated 10 HIV haplotypes calculated from reads mapping.

Haplotype	FJ061	FJ066	FJ065	FJ064	FJ061-h1
Abundance(%)	34.61	16.83	11.26	8.44	6.94
Haplotype	FJ061-h2	FJ066-h1	FJ066-h2	FJ065-h1	FJ065-h2
Abundance(%)	5.78	4.84	4.20	3.75	3.36

to map reads to a set of universal genes and infer the within-species strains abundances by utilizing single-nucleotide polymorphism (SNP) patterns. This tool again did not get enough mapped reads to report any strain abundances. Thus, we cannot report the results from StrainPhlAn or ConStrains.

4.3.1.3 Results for 10 HIV haplotypes

Haplotype abundances

Similar to 5 HIV haplotypes, by mapping simulated reads to the 10 HIV haplotypes by bowtie2, we were able to estimate the abundance for each haplotype as shown in Table 4.3.

Windows identification

Similar to 5 HIV haplotypes, the contigs alignment and windows identification was also applied on simulated contigs for 10 HIV haplotypes. VirBin generated 319 windows from all the contigs. Sorting these windows by their lengths, the top 3 are shown in Figure 4.4.

From the results, out of the top 100 windows, 36 contain 10 contigs, 34 contain 9 contigs, and 20 contain 8 contigs, 4 of the left windows contain contigs number greater than 10 and 6 contain contigs number less than 8. Out of the top 50 windows, 26 have 10 contigs, 16 contain 9 contigs, and 6 contain 8 contigs. Out of the top 25 windows, 15 contain 10 contigs, 8 contain 9 contigs, and 1 contains 8 contigs. Therefore, the haplotype number 10 can still be correctly figured out by the heights of most abundance windows.

Similar to 5 HIV haplotypes, the relative abundances for contigs within windows can be calcu-

lated by aligning reads to contigs with bowtie2. The relative abundances for contigs in the top 3 windows are shown in Figure 4.4.

```
>Window reference_contig: FJ066-h2_2_8913_9640  10.0    1555    3111    FJ065-h1        0.0416168598306
FJ061_1_15_4727 0.3479  1118059.0       FJ061   0.326607244946
FJ066_1_202_5355        0.1753  563200.0        FJ066   0.154240943818
FJ065_2_1171_3192       0.1072  344945.0        FJ065   0.104100618471
FJ064_1_393_9239        0.0851  273849.0        FJ064   0.0852086689881
FJ061-h1_1_263_3129     0.065   208688.0        FJ061-h1        0.078166415467
FJ061-h2_2_1006_6705    0.0585  187824.0        FJ061-h2        0.0673951457824
FJ066-h1_1_316_8981     0.0486  156274.0        FJ066-h1        0.0544150511928
FJ066-h2_1_1560_8705    0.0413  132928.0        FJ066-h2        0.049893456681
FJ065-h2_1_691_9145     0.0364  117390.0        FJ065-h2        0.0383555948236
FJ065-h1_1_373_8661     0.0362  116405.0        FJ065-h1        0.0416168598306
>Window reference_contig: FJ066-h1_1_316_8981   10.0    158     1493    FJ065-h2        0.0383555948236 (
FJ061_2_5197_9265       0.35    977042.0        FJ061   0.326607244946
FJ066_2_5339_7628       0.1539  430335.0        FJ066   0.154240943818
FJ065_3_3121_7669       0.1119  313399.0        FJ065   0.104100618471
FJ064_1_393_9239        0.0871  243864.0        FJ064   0.0852086689881
FJ061-h1_2_3843_9776    0.0672  187587.0        FJ061-h1        0.078166415467
FJ061-h2_2_1006_6705    0.0607  168648.0        FJ061-h2        0.0673951457824
FJ066-h1_1_316_8981     0.0558  156716.0        FJ066-h1        0.0544150511928
FJ066-h2_1_1560_8705    0.0449  126075.0        FJ066-h2        0.049893456681
FJ065-h1_1_373_8661     0.0386  108198.0        FJ065-h1        0.0416168598306
FJ065-h2_1_691_9145     0.0329  92061.0 FJ065-h2        0.0383555948236
>Window reference_contig: FJ066-h1_1_316_8981   10.0    1509    2444    FJ065-h1        0.0416168598306
FJ061_2_5197_9265       0.3548  690338.0        FJ061   0.326607244946
FJ066_2_5339_7628       0.1531  299044.0        FJ066   0.154240943818
FJ065_3_3121_7669       0.1074  209787.0        FJ065   0.104100618471
FJ064_1_393_9239        0.086   167954.0        FJ064   0.0852086689881
FJ061-h1_2_3843_9776    0.0674  131060.0        FJ061-h1        0.078166415467
FJ066-h2_1_1560_8705    0.0627  122278.0        FJ066-h2        0.049893456681
FJ066-h1_1_316_8981     0.0506  98033.0 FJ066-h1        0.0544150511928
FJ061-h2_3_6690_7641    0.0468  91557.0 FJ061-h2        0.0673951457824
FJ065-h2_1_691_9145     0.0395  76640.0 FJ065-h2        0.0383555948236
FJ065-h1_1_373_8661     0.0332  64746.0 FJ065-h1        0.0416168598306
```

Figure 4.4: Top 3 windows output from simulated contigs alignment for 10 HIV haplotypes. The output format are same as Figure 4.3.

EM clustering

With the identified windows and relative abundances, VirBin was also applied to cluster contigs into 10 groups for 10 HIV haplotypes. The evaluation results for clustering are shown in Table 4.4.

The results were also benchmarked with MaxBin. Similar to simulated 5 HIV haplotypes, 10 seed contigs from each haplotype were randomly selected and provided to MaxBin. It classified 25 out of 34 contigs, with 9 unclassified. The results are shown in Table 4.4. While MaxBin correctly classified one contig to FJ065-h1 and one to FJ066, most of the other contigs were not

Table 4.4: EM clustering results on simulated 10 HIV haplotype contigs for VirBin and MaxBin.

	virbin		MaxBin	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
FJ064	73.0	98.0	0.0	0.0
FJ061	100.0	80.0	0.0	0.0
FJ061-h1	98.1	88.1	0.0	0.0
FJ061-h2	99.5	89.2	0.0	0.0
FJ065	86.4	61.5	0.0	0.0
FJ065-h1	18.6	10.7	17.6	100.0
FJ065-h2	42.1	94.9	0.0	0.0
FJ066	84.6	95.8	43.4	69.2
FJ066-h1	90.9	79.2	0.0	0.0
FJ066-h2	60.0	47.6	0.0	0.0

appropriately clustered. We again tried StrainPhlAn and ConStrains on this simulated data set, but still, no reads can be mapped to available reference genes.

4.3.2 HIV real MiSeq data set

4.3.2.1 HIV real data set and contigs

To further assess the performance of the binning method, we applied VirBin on the real HIV quasispecies data set (SRR961514), sequenced from the mix of five HIV-1 (89.6, HXB2, JRCSF, NL43, YU2) strains with Illumina MiSeq sequencing technology [39]. This data set contains 714,994 pairs (2x250 bp) of reads that cover the five strains to 20,000x. The raw data set was pre-processed with FaQCs/1.3 [87] and Trimmomatic [15] to trim and filter low-quality reads or adapters. The left reads were then error-corrected with Karect [2]. After pre-processing, 774,044 reads were left.

By mapping pre-processed reads to the available 5 reference genomes by bowtie2, we were able to estimate the abundance for each haplotype as shown in Table 4.5.

We use the contigs assembled by PEHaplo as input for VirBin. PEHaplo produced 24 contigs

Table 4.5: Haplotype abundances calculated from reads mapping.

Haplotype	JRCSF	NL43	89.6	YU2	HXB2
Abundance (%)	29.79	25.18	21.68	12.62	10.87

from the real MiSeq HIV data set that can cover over 92% of the five HIV-1 strains. These contigs have a N50 value of 2,223 bp and the longest contig is 9,133 bp.

4.3.2.2 Results for HIV real data set

Windows identification

After aligning contigs to each other by blast, VirBin was applied on the aligned profiles and generated 197 windows. Sorting these windows by length, the top 5 are shown in Figure 4.5.

From the results, out of the top 100 windows, 44 contain 5 contigs, 39 contain 6 contigs, and 5 contain 4 contigs. Out of the top 50 windows, 27 contain 5 contigs, 16 contain 6 contigs, and 2 contain 4 contigs. Out of the top 25 windows, 17 contain 5 contigs, 5 contain 6 contigs, and 1 contains 4 contigs. Similar to the simulated haplotypes results, the haplotype number 5 can be estimated from the heights of most abundant windows.

While the windows identified on simulated contigs tend to have windows heights less than the haplotype number, the windows produced on contigs assembled have a lot of heights greater than the haplotype number. One reason could be that there may be "chimeric contigs" assembled, where a contig is joined by sequences from two or more haplotypes. The chimeric contigs can align with those correct contigs, thus increasing the height of the windows. The simulated contigs are error-free but not necessarily be complete due to the randomness. Therefore, there are a lot of windows with a height less than the haplotype number. Another possible reason is that there are repeated sequences for HIV real haplotypes. Let us denote a substring of a viral haplotype X as $X[start\_position, end\_position]$. The blast results between these HIV haplotypes show that

```
>Window reference_contig: 3077|1031     5.0     3       1640    YU2     0.1262
14712|1167      0.3013  9840417.0       JRCSF   0.2979
19109|4455      0.2079  6797490.0       89.6    0.2168
3077|1031       0.1914  6125241.0       HXB2    0.1087
17123|944       0.1753  5788432.0       NL43    0.2518
1985|810        0.1289  4221446.0       YU2     0.1262
>Window reference_contig: 1985|810      5.0     367     1993    YU2     0.1262
14712|1167      0.2961  9773461.0       JRCSF   0.2979
19109|4455      0.2098  6803613.0       89.6    0.2168
3077|1031       0.193   6129357.0       HXB2    0.1087
17123|944       0.1758  5795251.0       NL43    0.2518
1985|810        0.1292  4225063.0       YU2     0.1262
>Window reference_contig: 19109|4455    5.0     275     1148    HXB2    0.1087
14402|812       0.2749  6745597.0       NL43    0.2518
6779|3607       0.2415  5941781.0       JRCSF   0.2979
19109|4455      0.1982  4896019.0       89.6    0.2168
9615|725        0.1562  3843177.0       YU2     0.1262
1206|773        0.1317  3239063.0       HXB2    0.1087
>Window reference_contig: 19109|4455    5.0     361     1988    YU2     0.1262
14712|1167      0.2958  9768538.0       JRCSF   0.2979
19109|4455      0.21    6810527.0       89.6    0.2168
3077|1031       0.1932  6136343.0       HXB2    0.1087
17123|944       0.1757  5794854.0       NL43    0.2518
1985|810        0.1292  4225369.0       YU2     0.1262
>Window reference_contig: 3077|1031     5.0     540     1413    HXB2    0.1087
14402|812       0.2749  6745597.0       NL43    0.2518
6779|3607       0.2415  5941781.0       JRCSF   0.2979
19109|4455      0.1982  4896019.0       89.6    0.2168
9615|725        0.1562  3843177.0       YU2     0.1262
1206|773        0.1317  3239063.0       HXB2    0.1087
```

Figure 4.5: Top 5 windows output for contigs assembled from HIV real MiSeq data. The output format are same as Figure 4.3.

Table 4.6: EM clustering results on assembled 5 real haplotype contigs for VirBin and MaxBin.

	VirBin		MaxBin	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
HXB2	48.5	70.2	39.1	27.0
YU2	34.0	30.0	56.6	50.9
89.6	58.0	56.5	0.0	0.0
NL43	18.5	17.4	9.6	21.6
JRCSF	65.1	50.8	0.0	0.0

Table 4.7: EM clustering results on assembled 5 real haplotype contigs for VirBin and MaxBin.

	VirBin		MaxBin	
	Precision (%)	Recall (%)	Precision (%)	Recall (%)
HXB2	70.0	87.5	33.3	25.0
YU2	25.0	20.0	33.3	33.3
89.6	33.3	100.0	0.0	0.0
NL43	66.7	28.6	25.0	20.0
JRCSF	33.3	33.3	0.0	0.0

89.6[9046, 9669] can be aligned with JRCSF[1, 630] with an identity of 92%. Therefore, the alignment between contigs may not mean that they come from the same location of the consensus genome sequence, which can lead to an overestimation of the height of the windows.

EM results The similar clustering procedures were applied on identified windows by VirBin with relative abundances. MaxBin was run again as Benchmark The results from both tools are shown in Table 4.6. StrainPhlAn and ConStrains were applied on this real HIV data set too. StrainPhlAn was able to identify the HIV species, but could not report any strain information. ConStrains could not align enough reads to marker genes for further reporting strain abundances.

VirBiné results are not as good as for simulated HIV data sets. One of the reason may be the existence of imperfect assembled contigs, which may lead to the incorrect calculation of abundance levels and finally affect the clustering.

4.4 Discussion and Conclusion

Within the same viral quasispecies, binning the contigs from different haplotypes with k-mer frequencies (such as tetranucleotide frequencies) is not applicable because of the high sequence similarities between them. The only possible information to differentiate these contigs is their abundances inferred from reads coverage levels by mapping. However, the inhomogeneous reads coverage along the viral genome and close abundances between viral strains increase the difficulty to directly applying coverage levels for binning viral contigs. In this Chapter, we proposed a novel method to more accurately calculate the relative abundances for sub-contigs within windows. Moreover, the number of haplotypes in the quasispecies can be estimated from the consensus height of windows.

We have shown the utility of our tool on two simulated and one real viral quasispecies data sets, and benchmarked the results with MaxBin. The success of this method relies on the quality of input contigs and the abundance difference between viral haplotypes. When the assembled contigs can cover the most part of viral strains, the number of haplotypes in the quasispecies can be accurately identified. The empirical experience shows that it is difficult to classify two viral strains when the abundance difference between them is below 3%.

Chapter 5

Studying transcriptional regulations of miRNA genes with Cap-seq

5.1 Introduction

MicroRNAs (miRNAs) are a large family of ∼21 nucleotide-long RNAs that have been uncovered as key regulators of gene expression at post-transcriptional level in metazoans, plants, and viruses [63, 66, 10]. In metazoans, mature miRNAs and argonaute (AGO) proteins form into the miRNA-induced silencing complex (miRISC), within which miRNAs base-pair to the 3'-UTR of target mRNAs and inhibit protein synthesis by either repressing translation or promoting mRNA degradation. It was inferred that more than one-third of all protein-coding genes are regulated by miRNAs [94]. miRNAs have also been discovered to play a crucial role in precision medicine. Precision medicine attempts to characterize the genetic background of patients and classify them into subpopulations that differ in their susceptibility to a particular disease [30, 58, 91]. The capability of modulating a vast number of protein-coding genes makes miRNA powerful regulators of the different cellular processes involved in the pathogenesis of various types of diseases, including cardiovascular diseases and cancer. For example, liver miRNA miR-122, as the most abundant and most specific liver miRNA, is most likely to represent a novel biomarker for cardiovascular and metabolic diseases as it plays a central role in lipid and glucose homeostasis and is detectable in serum and plasma [156]. Differential expression of miRNAs have also been observed in tumor

tissues. Their alteration expression in prostate cancer has been well documented [96]. Because of their important regulatory functions, many studies have focused on miRNA annotation and identifying their targets [32, 164, 78]. However, how miRNAs themselves are expressed and regulated is not fully understood.

In the canonical miRNA biogenesis pathway, miRNAs are processed from longer transcripts, which are referred to as primary miRNAs (pri-miRNAs) [66]. Pri-miRNAs are either transcribed by polymerase II (Pol II) from independent genes or derived from the introns of protein-coding genes [77, 16]. Two members of the RNase III family of enzymes, Drosha and Dicer, further process pri-miRNAs to mature miRNAs [76, 28, 68]. First, Drosha cleaves the hairpin structure of a pri-miRNA to an $\sim$70-nucleotide precursor miRNA (pre-miRNA) in the nucleus. Pre-miRNAs are then exported to the cytoplasm by XPO5, where Dicer cleaves off the loop region of the hairpin and further processes it to $\sim$21-bp mature miRNA(s). Recent studies have uncovered several non-canonical ways of generating miRNAs, demonstrating the complexity of miRNA biogenesis. One class of unconventional miRNAs is called mirtrons, which are encoded in introns, bypass Drosha processor but rely on splicing machinery for pre-miRNA generation [11, 130]. miRNAs in mammals have been shown to frequently utilize alternative promoters in different cell types, and pri-miRNAs may encode subsets of clustered miRNAs [24]. Pri-miRNA transcripts can be cleaved by cytoplasmic Drosha in human cells [35]. Another study on mice has uncovered a second class of non-canonical miRNAs, of which the pre-miRNAs are 5'-capped and generated directly by transcription [161].

Although the genomic coordinates of mature and precursor miRNAs have been annotated in databases such as miRBase [48], very little is known about the coordinates of pri-miRNAs. RNA-seq technology [155] has been proved as an efficient way to annotate protein-coding genes. Mature mRNAs contain a 5' 7-methylguanosine (m^7G) cap and a long 3' polyadenylated (poly(A)) tail

and are relatively stable, so they can be well extracted from cells and sequenced. The sequenced RNA fragments are then mapped to the reference genome for gene annotation. However, since the original 5' ends of primary miRNA transcripts are rapidly cleaved off by Drosha during miRNA maturation, regular RNA-seq technology cannot be used to find the primary TSSs of miRNA genes. Pri-miRNAs are usually transcribed by Pol II and also contain a 5' m^7G cap and a 3' poly(A) tail [77], indicating that the biological features related to Pol II transcription can be used to identify the transcription initiation sites for miRNA genes.

To identify the primary TSSs of miRNAs, some computational methods have been implemented based on features related to Pol II transcribed genes, such as transcription factor binding sites (TFBSs), Pol II binding, and chromatin states including histone modifications and nucleosome positioning [132, 111, 31, 29]. Typically, Pol II and H3K4me3 are highly enriched at active promoters, while nucleosomes are depleted at the TSSs. Wang et al. [154] designed a statistical model to mimic Pol II binding patterns at the promoters of highly expressed protein-coding genes and used it to search for similar Pol II binding patterns upstream of all intergenic miRNAs in human breast cancer cells to identify primary promoters. They verified their findings by checking the conservation, CpG content, and activating histone marks in the identified promoter regions. Ozsolak et al. [111] combined nucleosome mapping with ChIP-chip screens for H3K4me3, H3K9/14ac, Pol II and Pol III signatures to identify the proximal promoter regions of pri-miRNAs in human genome. They tested their algorithm on human annotated protein-coding genes and predicted the transcription initiation regions to a resolution of 150 bp. With the same method, the transcription initiation regions of 175 transcriptionally active miRNAs were determined. Saini et al. [131] predicted the 5' ends of intergenic pri-miRNAs in human, mouse and rat genomes by combining the features of TSSs predictions, CpG islands and 5' cap analysis of gene expression (CAGE) tags. miRStart [29] built a SVM model using the features of CAGE tags, TSS Seq libraries and

H3K4me3 chromatin signature from ChIP-seq to identify the TSSs of human miRNAs. The model was trained on 7,268 protein-coding genes with unique TSS and identified 847 putative TSSs for the 940 human pre-miRNAs obtained from miRBase.

While the methods discussed above [154, 111, 131] have predicted TSSs for miRNA genes in mammal genomes, their prediction results have low resolution (hundreds of bps) because the typical distribution patterns of Pol II and chromatin features surrounding promoters may not hold for any particular gene. Even for some actively transcribed genes (29/85), the distance between the TSS and the closest Pol II peak can be over 1000 bp (Figure S1(B)). A more accurate method is to take advantage of the cap structure at 5' ends of Pol II transcribed RNAs. Mapping the capped sequences to reference genomes will enable direct discovery of the TSSs. CAGE and Cap-seq have been used to directly sequence RNAs with 5' m^7G caps, which are used to identify the candidate TSSs. CAGE [65] uses a so-called "cap trapper" method to capture full-length mRNAs and sequence the 5' ends with Sanger sequencing technology. However, CAGE is not widely used to map TSSs for each gene because of the cost and sequencing depth. With the development of high throughput sequencing technology, enriching capped RNA transcripts followed by next-generation sequencing (NGS) technology (Cap-seq or deepCAGE [38]) has been used to sequence the capped RNAs in the whole genome.

The mouse is a popular mammalian model system for genetic research, for which some Cap-seq datasets have been generated [161, 49]. Recently the Cap-seq study in mice [161] has uncovered a non-canonical way of generating pre-miRNAs, in which the pre-miRNAs are generated directly by transcription and their 5' ends are m^7G capped. These 5' m^7G capped pre-miRNAs prefer to be exported from the nucleus to the cytoplasm by exportin 1 (XPO1) and after Dicer processing, only 3p-miRNA is efficiently loaded onto the AGO complex. This special class of 5'-capped pre-miRNAs have also been discovered in the human genome (miR-320a) [161], but whether they also

exist in other non-mammalian species is still unknown.

Caenorhabditis elegans (*C. elegans*) is also a well established model organism for genomic studies. The worm is a simple multicellular organism but with a variety of tissue types and a short life cycle [33]. Therefore, many functional genomic sequencing datasets, including Cap-seq [49, 27, 67], have been generated on this species. The transcription regulation in this animal is quite different from that in mammals. For example, the primary transcripts of about 70% of its protein-coding genes undergo trans-splicing [144]; and its pri-miRNA transcripts are exported by XPO1 and possibly processed in nuclear pore [19].

In this study, we utilized available Cap-seq datasets to study the transcription regulation of miRNAs in *C. elegans* and mouse. The main results are summarized below.

- We identified a group of candidate 5' m^7G capped pre-miRNAs in *C. elegans*.
- We classified another class of miRNAs with non-canonical transcription mechanisms, for which the pre-miRNAs may be generated by both the canonical miRNA pathway with Drosha and the non-canonical pathway without Drosha.
- Based on the capping signals for miRNA genes in clusters, we proposed a hypothesis that these pri-miRNA transcripts might undergo cytoplasmic re-capping during the pre-miRNAs generation process.
- We developed a method to separate these identified primary miRNA promoters as broad or divergent and characterized them by analyzing the H3K4me3 and Pol II binding surrounding them.

5.2 Materials and Methods

5.2.1 Datasets and processing

The small RNA Cap-seq data for new born mouse was retrieved from the study by Xie et al. [161], and was downloaded from Sequence Read Archive (SRA) with run number SRR1022391. The small RNA Cap-seq data for mixed staged embryos of *C. elegans* is from the study by Chen et al. [27], and was downloaded from NCBI Gene Expression Omnibus (GEO) with accession number GSE42819. The small RNA-seq data of *C. elegans* L4 stage was also downloaded from GEO database with accession number GSM916519. The ChIP-seq data of H3K4me3 and Pol II for *C. elegans* were also obtained from NCBI GEO database, with accession number GSE28770 and GSE15535, respectively. Small RNA-seq data for detecting mature miRNAs in *C. elegans* was downloaded from GEO with accession number GSM916519.

Raw sequencing data sets are in SRA format and were dumped to FASTQ format by SRA Toolkit [145]. The FASTQ files were then mapped to *C. elegans* (WBcel235) and Mouse (GRCm38) reference genome using bowtie [72] allowing 2 mismatches and 3 mismatches for reads length of 36 nt and 50nt, respectively. Only uniquely mapped reads were reported in the output SAM files and were visualized by GenomeView [1].

5.2.2 Clustering of 5' end reads

Small RNA Cap-seq data for *C. elegans* and mouse are strand specific. Mapped reads on forward and reverse strand were analyzed independently. We identified the transcription initiation clusters (TICs) with similar methods from the study by Chen, et al [27]. First, mapped reads with same strand and 5' end positions were combined and denoted as cap-stacks. Second, all cap-stacks containing five or more tags were clustered using a single-linkage approach: two or more stacks

were clustered together if the distance between two adjacent stacks is less or equal to 50 bp. The position covered by the most 5' ends within the TIC was defined as the mode, which represents the TSS for the TIC. In the case of two or more positions with the same number of tags, the one furthest upstream was selected as the mode. Here in total we obtained 32,530 TICs in *C. elegans* and 4,903 TICs on mouse chromosome 7.

5.2.3 Statistical analysis

Because the Cap-seq is not perfect and may involve contamination or artifacts, we used a Poisson distribution to model the background noise following the previous work [168]. Those TICs that are significantly enriched with sequencing reads are reported (Poisson distribution p-value based on λ). Since the reads of Cap-seq are not randomly distributed along the genome, we estimate a dynamic parameter λ_{local}, defined for each TIC as:

$$\lambda_{\text{local}} = \min[\lambda_{5\text{k}}, \lambda_{10\text{k}}] \tag{5.1}$$

where $\lambda_{5\text{k}}$ and $\lambda_{10\text{k}}$ are estimated from the 5kb or 10kb window centered at the mode of the TIC. Our model is built on the assumption that a TIC is reliable if the number of reads enriched inside of the TIC is significantly higher (default with p-value $<10^{-5}$) than the number of reads located in the TIC when they are randomly distributed in the local region.

Results of Cap-seq and small RNA-seq reads mapped to tRNAs were subject to a two-tailed paired sample t-test to estimate their differences, with p-value <0.01 considered statistically significant. Both Cap-seq and small RNA-seq mapping results on tRNAs were normalized by their coverage depths on the genome. Since they only cover a small part of the whole genome, the coverage depth was calculated as the total number of reads mapped multiply the reads length and

divided by the length of covered genome region. The similar Poisson model is also used to estimate the enrichment of Cap-seq reads on tRNAs.

5.2.4 miRNA cluster and intergenic pre-miRNAs identification

miRNA clusters were retrieved from miRBase with a distance threshold of 1000 bp. The distance between pre-miRNAs in mice are usually longer than 1000 bp. Thus, we did not find any miRNA cluster in mice with our threshold.

To identify intergenic miRNAs from *C. elegans* and mouse genome, we downloaded the miRNA annotations from miRBase Release 21 and gene annotaion gff3 files. The gene annotation files for *C. elegans* and mouse were downloaded from Ensembl website (http://www.ensembl.org/index.html). Pre-miRNAs from both miRBase and Ensembl gene annotation files were isolated and those miRNAs that are not covered by protein-coding genes, non-coding RNA genes, snoRNA genes and snRNA genes were identified as intergenic miRNAs. In the case that same pre-miRNA is annotated both in miRBase and gene annotation file, we only kept the one in miRBase. In total, we identified 134 intergenic miRNAs in *C. elegans* and 80 intergenic miRNAs on mouse chromosome 7. The flanking region upstream of the 5' end of intergenic pre-miRNA was also identified, and TICs detected in this region were annotated as the candidate primary TSSs for the miRNA. The TICs identified within the pre-miRNA region were annotated as the pre-cap TICs.

5.2.5 Finding bidirectional and multiple TSSs promoters

With the capped RNA reads, we identified transcription initiation sites and annotated their promoters as bidirectional or broad promoters (promoters with multiple TSSs) in *C. elegans*. We used the identified TICs to annotate these promoters. First for each TIC (*A* in Figure 5.7 (B))on the

plus strand, the most close downstream (*C* in Figure 5.7 (B)) and upstream (*B* in Figure 5.7 (B)) minus strand TICs were searched. If the distance between the upstream minus TIC (*B*) and the plus strand TIC (*A*) is less than threshold (here we use 300bp), the two TICs were treated as from the same promoter and the promoter was annotated as bidirectional. The downstream minus strand capped peak (*C*) acts as the boundary for detecting multiple transcripts. All the plus strand TICs upstream of it were annotated as from the same promoter. To identify the multiple transcripts on the minus strand, the most close upstream plus strand TIC (*D*) of *B* was found and all the minus strand TICs between *D* and *B* were annotated as from this promoter. Those left minus strand TICs were annotated with the similar method.

With this approach, we detected 11,272 promoters in *C. elegans*, of which, 6,149 are bidirectional promoters and 2,359 are broad promoters. The most upstream 5' ends of both plus and minus strand TICs were used to represent for the TSSs of these promoters. Only one TSS on each strand was selected as representing TSS for one promoter.

5.3 Results

5.3.1 Identification of primary miRNA TSSs in *C. elegans* and mouse

5.3.1.1 Overview of primary miRNA TSSs annotation

We used the single-linkage clustering method to detect transcription initiation clusters (TICs) [27] from Cap-seq data in *C. elegans* and mice. For each TIC, we also used a Poisson distribution to model the local background noise and test whether it is significantly enriched with Cap-seq reads by calculating a *p*-value (see Materials and Methods). The intronic miRNAs are usually processed as part of their host-gene mRNA [10] and thus their transcriptions coordinate with the

protein-coding genes. Therefore, in this study, we have focused on identifying the primary TSSs for intergenic miRNAs. In both species, we first identified the intergenic miRNAs that are not covered by protein-coding genes, non-coding RNA genes, small nucleolar RNA (snoRNA) genes, or small nuclear RNA (snRNA) genes. Then the flanking region between 5' end of pre-miRNA and the closest upstream gene was searched for TICs. The identified TICs were annotated as candidate *primary TICs* (Figure 5.1). The region within pre-miRNA was also searched for TICs, and the identified TICs were annotated as *pre-cap TICs* (Figure 5.1). Moreover, we also searched for miRNA clusters from miRBase with a distance threshold of 1000 bp. For miRNA clusters, primary TSS(s) are annotated as the TIC(s) upstream of the 5' end of the first pre-miRNA in the cluster and pre-cap TIC(s) are annotated as the TIC(s) within the pre-miRNAs in the cluster.

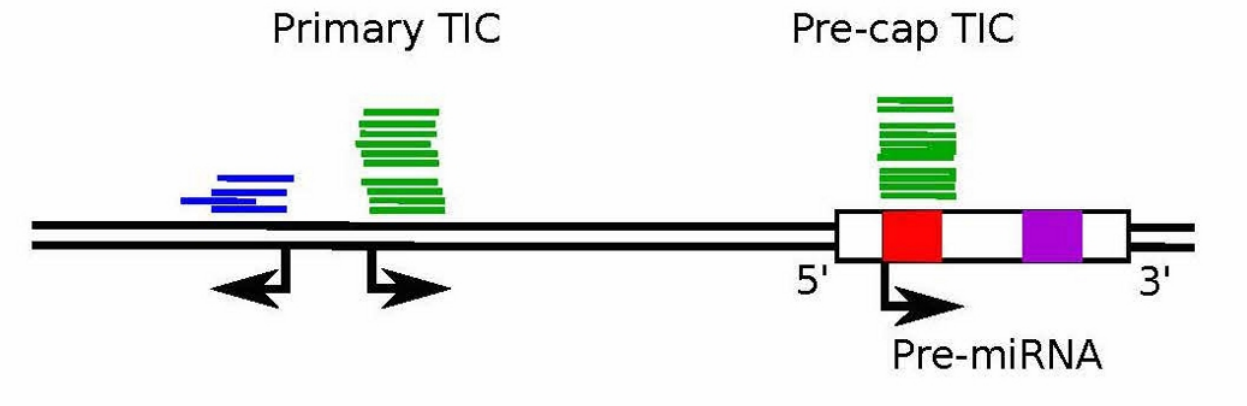

Figure 5.1: Primary TIC is located upstream of the pre-miRNA, while pre-cap TIC is located inside of the pre-miRNA. The annotation of a pre-miRNA is usually a stem-loop that includes the pre-miRNA and the lower stems. However, the real pre-miRNA only includes the red, purple sequences (mature miRNAs) and the loop between them. Therefore, the pre-cap TIC starts from the 5' end of a 5p-miRNA. Each blue or green bar corresponds to a mapped read, where green indicates the plus strand and blue the minus strand. The Cap-seq datasets were sequenced in a strand-specific way. Only the reads on the same strand of the miRNA are considered.

In *C. elegans*, we identified 134 intergenic miRNAs and 16 miRNA clusters. With the retrieved Cap-seq data [27], 70 intergenic miRNAs were identified with at least one candidate primary TIC

in the flanking regions, and 9 miRNAs were identified with at least one pre-cap TIC inside of the precursors. For those miRNAs with candidate primary TICs or pre-cap TICs, 8 were identified with both of them. The modes (highest coverage of reads' 5' ends) of the primary TICs are used to represent the candidate TSSs of miRNAs. In the 16 miRNA clusters, 6 were found to contain both primary TSSs and pre-cap TICs, and 2 clusters were found to contain only primary TSSs (Table S1).

In mice, we used the Cap-seq data from the study by Xie et al. [161]. They performed Cap-seq to find the unconventional pre-miRNAs whose 5' ends are generated directly by transcription initiation with Pol II and thus 5' m^7G capped. Based on their results, there are the highest number of 5'-capped pre-miRNAs on chromosome 7, so we focused on identifying the primary TSSs for miRNAs within mouse chromosome 7. We identified 80 intergenic miRNAs on this chromosome, of which 10 miRNAs were identified with at least one pre-cap TIC and 37 miRNAs were identified with at least one candidate primary TICs. Of those miRNAs with candidate primary TICs or pre-cap TICs, 2 were found to contain both of them (Table S2).

5.3.1.2 Comparison with previous work

In the previous study [49], Gu et al. developed the CapSeq protocol to enrich and sequence longer (70-90 nt) 5'-capped RNA transcripts, and CIP-TAP cloning to isolate and sequence 5'-capped small (18-40 nt) RNAs. They applied these two 5' anchored RNA deep-sequencing approaches onto *C. elegans* and mouse genomes and annotated the primary TSSs for miRNAs in both of them. As a result, they identified at least one TSS for 55 individual pre-miRNAs and 9 miRNA clusters in *C. elegans*, and 134 individual pre-miRNAs in mice, with 7 miRNAs annotated on chromosome 7. Comparing with their results, we identified the primary TSSs for 43 additional miRNAs, 2 additional miRNA clusters in *C. elegans* and 37 additional miRNAs on mouse chromosome 7

(Table S3).

In another study [67], Kruesi et al. devised a global run-on cap sequencing (GRO-cap) method to capture and sequence only those 5' m^7G capped RNAs in *C. elegans* embryos, starved L1 larvae, and L3 larvae. With the GRO-cap sequencing data, they annotated the primary TSSs for 52 individual pre-miRNAs and 5 miRNA clusters. We identified similar primary TSSs for those overlapping pre-miRNAs (24 miRNAs). Furthermore, we annotated the TSSs for 46 more individual miRNAs and 3 more miRNA clusters (Table S4).

5.3.2 5' m^7G capped pre-miRNAs are identified in *C. elegans*

As shown in Figure 5.1, reads in pre-cap TICs are usually aligned very well with the 5' ends of pre-miRNAs. There is a possibility that these reads were actually sequenced from uncapped pre-miRNAs rather than capped RNAs. In this section, we first investigate whether usually uncapped pre-miRNAs are highly enriched by Cap-seq protocol.

5.3.2.1 Possible 5' recessed RNAs enriched by Cap-seq

To enrich for capped RNA in Cap-seq experiments for *C. elegans*, exonuclease Terminator and calf intestinal alkaline (CIP) were used to remove the uncapped RNAs [27]. However, some uncapped RNAs, such as pre-miRNAs and tRNAs, may not be accessed efficiently by Terminator/CIP because of their 5' recessed ends in their secondary structures. As a result, some of the Cap-seq reads in pre-cap TICs may actually come from pre-miRNAs. To investigate the contamination of pre-miRNA reads in Cap-seq data, we downloaded small RNA-seq data (GSM916519) for *C. elegans* and mapped the reads to the reference genome as the control. Those miRNAs that are detected as expressed by small RNA-seq data might be observed in Cap-seq data as well. We then quantified the number of Cap-seq reads aligned to those expressed pre-miRNAs (mapped with sufficient small

RNA-seq reads) (Figure S2(A)). The results showed that the numbers of Cap-seq reads mapped to pre-miRNAs are not proportional to the numbers of small RNA-seq reads mapped. For many highly expressed miRNAs (22/33), there are few or no Cap-seq reads aligned to their precursors, indicating that many Cap-seq reads may not be pre-miRNAs. In addition, pre-miRNAs, serving as an intermediate during miRNA maturation, are quickly processed by Dicer and thus tend not to be enriched by Cap-seq. Previous work has shown that the data of carefully designed pre-miRNA sequencing only contains less than 1% reads that can be mapped to pre-miRNAs [82].

According to Chen et al. [27], there was no step to remove tRNAs in the Cap-seq protocol. We then did the similar comparison for tRNAs because tRNAs may escape treatment of the Terminator and CIP for the same reason as pre-miRNAs. The results showed that, although almost all the annotated tRNA genes in the worm were expressed (604/605), most of these tRNAs (463/605) do not have any Cap-seq reads mapped (Figure S2(B)). A two tailed paired sample t-test was used to estimate the mean difference between the normalized Cap-seq reads and small RNA-seq reads mapped to tRNAs. We got a p-value as 1.12e-185, showing that the tRNA reads captured by two protocols are significantly different (alpha level as 0.01). Pearson's correlation and Spearman's rank correlation between two mapping results were also calculated. The correlation coefficient results (-0.042 and -0.125, respectively) suggest that they are poorly correlated. Most of those tRNAs mapped with Cap-seq reads have less than 10 reads (124/142, Figure S3), implying that these reads might be caused by random contamination. We then used the Poisson distribution to model the background noise (see Materials and Methods). 31 tRNAs were reported as significantly enriched with Cap-seq reads at the cutoff of 10^{-5}. Those tRNAs that are highly enriched for Cap-seq reads are usually overlapped with the repeat regions. Considering that most of the other tRNAs have few or no Cap-seq reads mapped, we infer that these regions might be transcribed by Pol II and produce capped RNAs under certain conditions.

We also suspected that some of the Cap-seq reads within pre-miRNAs might be mature miRNAs. However, mature miRNAs are short and usually do not possess complex secondary structures. They could be efficiently removed by Terminator/CIP treatment [44]. The deficit of Cap-seq reads on most mature miRNAs adds support to this. Hence the pre-cap TICs are not likely formed by mature miRNAs either. The Cap-seq study on mice have also shown that uncapped RNAs constitute less than 10% of the total sequenced reads [161]. Considering the above analysis together, we posited that although Cap-seq inevitably contains some uncapped RNAs, many of the reads mapped to pre-miRNAs are likely sequenced from capped RNAs.

5.3.2.2 Defining 5' m^7G capped pre-miRNAs with pre-cap TICs

RNAs synthesized by Pol II are 5' m^7G capped cotranscriptionally. A previous study [161] has documented a new class of unusual miRNAs in new-born mice, for which the 5' ends of the pre-miRNAs are m^7G capped and coincide with their TSSs. These miRNAs were suggested to be generated without Drosha processing, with their 5' ends determined directly by transcription initiation and the 3' ends generated by transcription termination.

To examine whether there are the same class of miRNAs in *C. elegans*, we analysed those pre-miRNAs with pre-cap TICs mapped inside. The 5' ends of pre-cap TICs are usually consistent with the 5' ends of pre-miRNAs or 5p-miRNAs. Therefore, those pre-miRNAs that were detected with pre-cap TICs should acquire m^7G cap at their 5' ends and were annotated as 5'-capped miRNAs. We also applied our method to mouse Cap-seq data obtained from the study by Xie et al. [161] and compared our results with theirs. Our results have uncovered all the 9 intergenic 5'-capped pre-miRNAs on mouse chromosome 7 as shown in the paper, validating our method. Besides, we also annotated the primary TSSs for 37 miRNAs with the same dataset. Strikingly, new candidate TSSs were also found upstream of two 5' capped pre-miRNAs (mmu-mir-344c and mir-344i). In

total, we identified 9 5'-capped miRNAs in *C. elegans* (Table S1) and 10 on mouse chromosome 7 (Table S2).

We also used statistical analysis to evaluate the enrichment of capped RNA reads on the 5' ends of pre-miRNAs. With the calculated *p*-values, 7/9 5'-capped pre-miRNAs in *C. elegans* and 9/10 in mice are significantly enriched with Cap-seq reads (p-value $<10^{-5}$; Table S1).

To look for sequence motifs surrounding these identified putative miRNA TSSs, we plotted the nucleotide composition around them by Weblogo [34]. A strong YR motif was observed at both the putative primary miRNA TSSs or pre-cap TSSs of independent miRNAs, in which Y represents pyrimidine, R represents purine and R locates at the TSSs (+1 position, Figure 5.2).

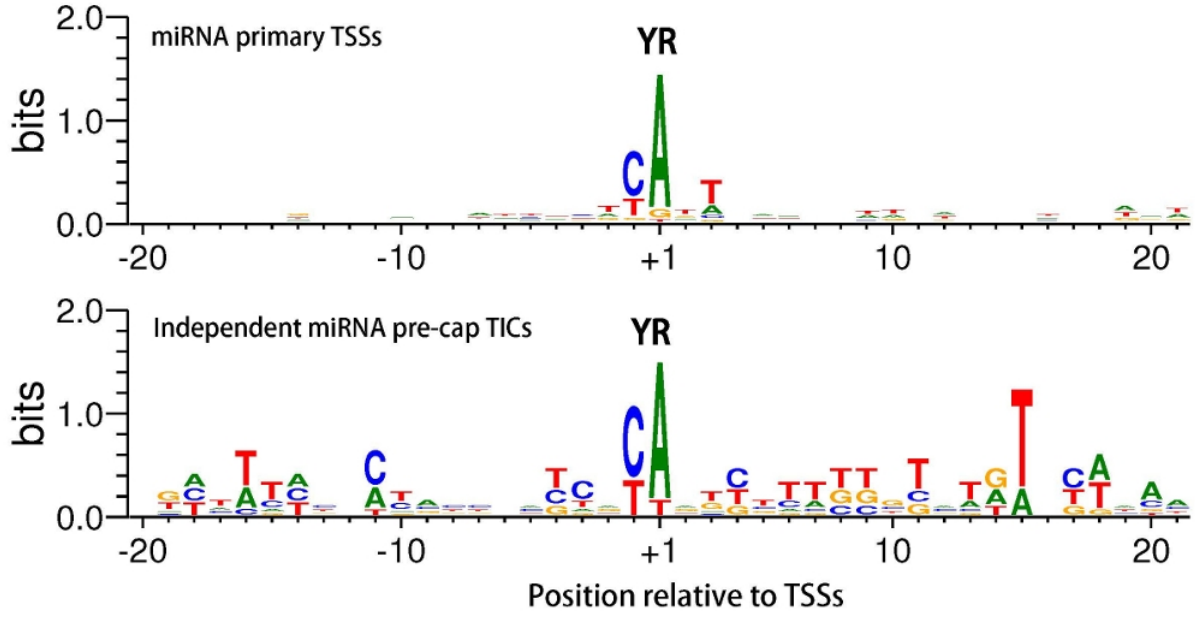

Figure 5.2: Sequence motif analysis of nucleotides around putative miRNA TSSs (+1). The nucleotide height (in bits) stands for the log_2 ratio of the observed nucleotides frequency relative to the background genomic nucleotide composition. The YR motif is observed at the putative primary miRNA TSSs and independent miRNA pre-cap TSSs.

5.3.3 M^7G capped pre-miRNAs often have upstream primary TICs

In both *C. elegans* and mice, we noticed that some of these 5'-capped pre-miRNAs also have primary TICs in the region from the pre-miRNA 5' end to the closest upstream gene. In *C. elegans*, 8 out of 9 5'-capped pre-miRNAs have been detected with primary TICs; while in mice, 2 out of 10 5'-capped pre-miRNAs have upstream primary TICs. These upstream TICs are usually not very far from the pre-miRNAs and the Pol II binding at these TICs elongate to the downstream miRNAs, indicating that they are connected with the miRNAs. To our knowledge, this phenomenon has not been described in other studies. Two examples in *C. elegans* are shown in Figure 5.3. Since pre-miRNAs can be generated both in the canonical way as from a long pri-miRNA by Drosha and in the non-canonical way by transcription initiation and termination, we ask which TIC corresponds to the real primary TSS of the miRNA or can both of them produce pre-miRNAs?

We proposed two possible explanations for this phenomenon. The first explanation is that there could be multiple isoforms for these miRNA genes. For example, the second discovered miRNA in *C. elegans* (let-7) has been detected with at least three primary transcripts [17]. It was also reported that genes often use alternative promoters in a developmental stage or cell type specific way that can spread up to thousands of kilobases. As an example, about one half of the protein-coding genes in human and mouse genomes have multiple alternative promoters [37]. Therefore, these miRNA genes may also contain mulitple alternative promoters that can generate several isoforms (Figure 5.4 (A)). The other primary TICs may produce longer miRNA transcripts that are subject to the canonical miRNA processing procedures involving Drosha. Since the 5'-capped pre-miRNAs are most likely generated directly by transcription, two paths may be able to lead to the maturation of the same miRNA: one with Drosha and the other without. Because we used the dataset from mixed-stage embryos, these miRNA genes may employ alternative promoters and produce diverse

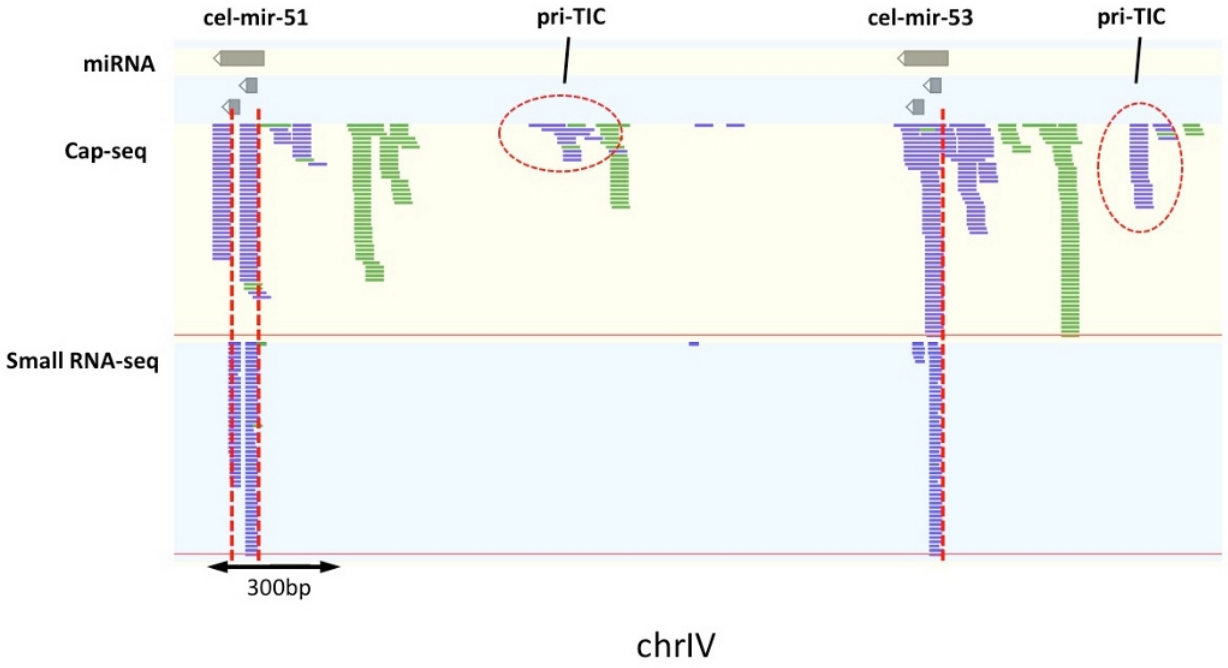

Figure 5.3: Primary TICs are detected upstream of 5'-capped miRNAs cel-mir-51 and cel-mir-53. Multiple Cap-seq peaks are observed in pre-miRNA regions. The mapped reads were visualized by GenomeView [1]. Each blue or green bar corresponds to a mapped read, where green indicates the plus strand and blue the minus strand. The reads in upper panel are from Cap-seq dataset, with uniform length of 36 nt. The reads in lower panel are from small RNA-seq dataset, with length in range from 14 nt to 26 nt.

isoforms at different stages. We also observed that for some miRNAs (Supplementary Figures S5 - S7: cel-mir-235, cel-mir-244, cel-mir-238, and cel-mir-228), the upstream TICs are very close to the pre-miRNA, likely that they are generated from the same promoter as the pre-cap TIC.

The second explanation is that the TICs upstream of pre-miRNA may be transcribed enhancers or promoters, which generate transcripts that will not produce miRNAs. Recently several studies have shown that promoter and enhancer regions can be transcribed in human, mouse and *C. elegans* genomes [40, 141, 27]. The transcription in promoters and enhancers are usually associated with downstream genes. It is suggested in *C. elegans* that the elongation from an upstream enhancer toward a downstream gene may have the potential to deliver Pol II to a proximal promoter, or al-

ternatively function directly as a distal promoter [27]. Thus, the upstream TICs may be transcribed in the enhancer regions and have a regulatory effect on the downstream miRNA genes (Figure 5.4 (B)).

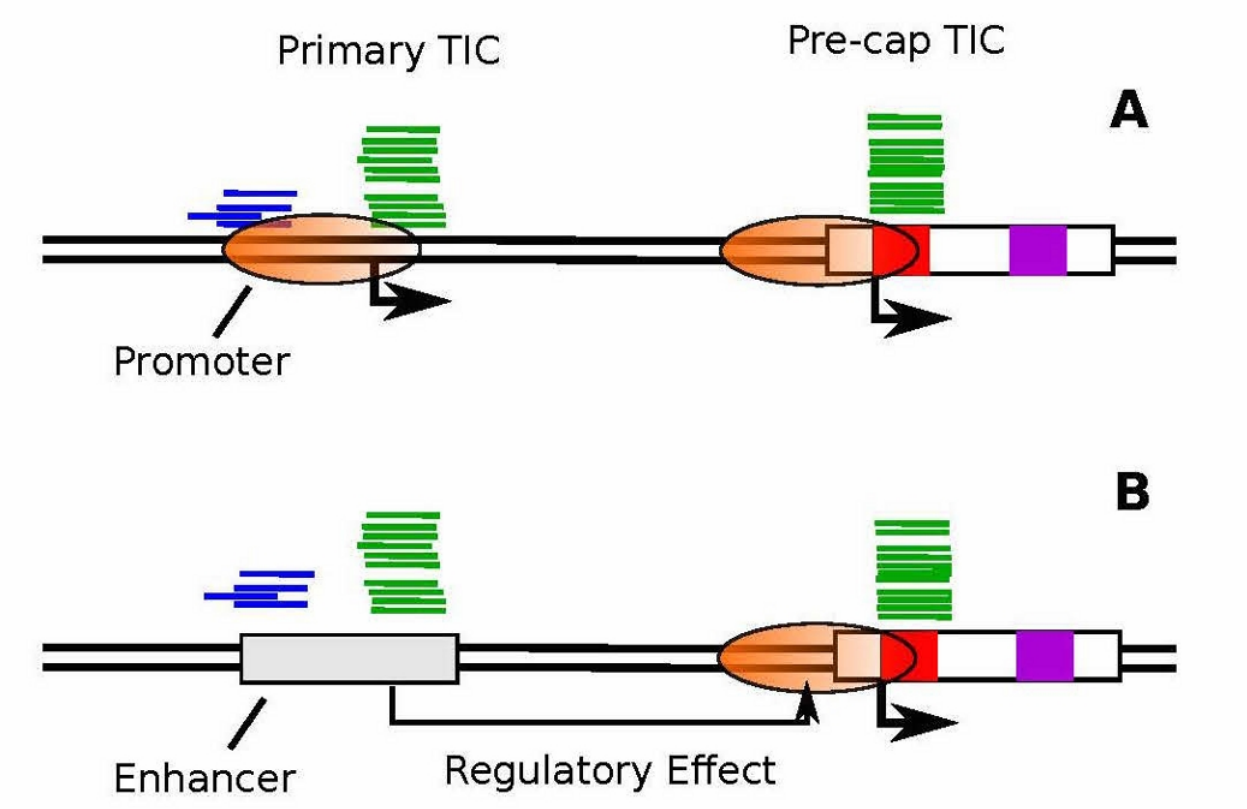

Figure 5.4: Upstream primary TICs also exist for m^7G capped pre-miRNAs. (A) Two alternative promoters for the same miRNA. Both of them are able to generate the transcripts that produce the same mature miRNA. (B) The miRNA transcript is generated by the pre-cap TIC. Upstream TIC(s) correspond to transcribed enhancer(s).

5.3.4 5p-miRNAs are produced from the identified m^7G capped pre-miRNAs

The m^7G capped pre-miRNAs have been reported to produce single 3p mature miRNAs in mice [161]. The main explanation is that the capped 5p-miRNA is not efficiently loaded onto Ago complex. However, we noticed that some identified 5'-capped pre-miRNAs in *C. elegans* also generate

5p mature miRNAs based on the annotations in miRBase (Table S1). While there is a possibility that some of the Cap-seq reads mapped to pre-miRNAs may be uncapped due to contamination or technical artifacts, most of these identified m^7G capped pre-miRNAs are significantly enriched with capped RNA reads according to our previous analysis. We surmise that there might be alternative pathways for producing 5p-miRNAs from those 5' m^7G capped pre-miRNAs. Similar observations were also made for identified 5'-capped pre-miRNAs in mice: four of them (mmu-mir-484, mmu-mir-1903, mmu-mir-344f and mmu-mir-344i) actually prefer to generate 5p mature miRNAs [161].

The 5p mature miRNAs could be generated by alternative primary TSSs. As many of these identified 5'-capped pre-miRNAs also have candidate primary TSSs, canonical primary miRNA transcripts may be generated from them and produce 5p-miRNAs. Studies have shown that mature miRNA selection from 5' and 3' strands of the same precursor is highly regulated and varies under different cell types, developmental stages and disease states [12, 97]. Capped 5p mature miRNAs may be produced under specific conditions to promote its target gene's expression.

5.3.5 Multiple transcription initiation sites for miRNA clusters in *C. elegans*

miRNAs in a cluster are close to each other, usually coexpressed and transcribed as a single pri-miRNA [74, 10]. Previously each miRNA cluster in *C. elegans* has been annotated with one primary TSS using Cap-seq [49] or GRO-cap [67] datasets. Strikingly, the datasets we used here have shown that the TICs for miRNA clusters in *C. elegans* can have a broad distribution with multiple strong peaks across the whole cluster. We identified primary TICs for 8 clusters, of which 7 have TICs inside of the clusters. One example of cluster cel-mir-35-41 is shown in Figure 5.5. The similar phenomenon is also observed in individual pre-miRNAs, as shown in Figure 5.3, the Cap-seq signal within cel-mir-51 and cel-mir-53 also displays multiple strong peaks. We noticed

that these capped reads peaks inside of clusters were located on both arms of pre-miRNAs, with 5p-peaks have the same 5' ends of 5p-miRNAs and 3p-peaks start from the 3' ends of 3p-miRNAs (Figure 5.5). As analyzed above, not many pre-miRNAs or mature miRNAs were kept in the Cap-seq experiments and many of the reads mapped are likely to be capped RNAs. This is further supported by the coordinate differences between the capped peaks and the miRNA/miRNA* on 3p arms. We proposed three hypotheses to explain this phenomenon.

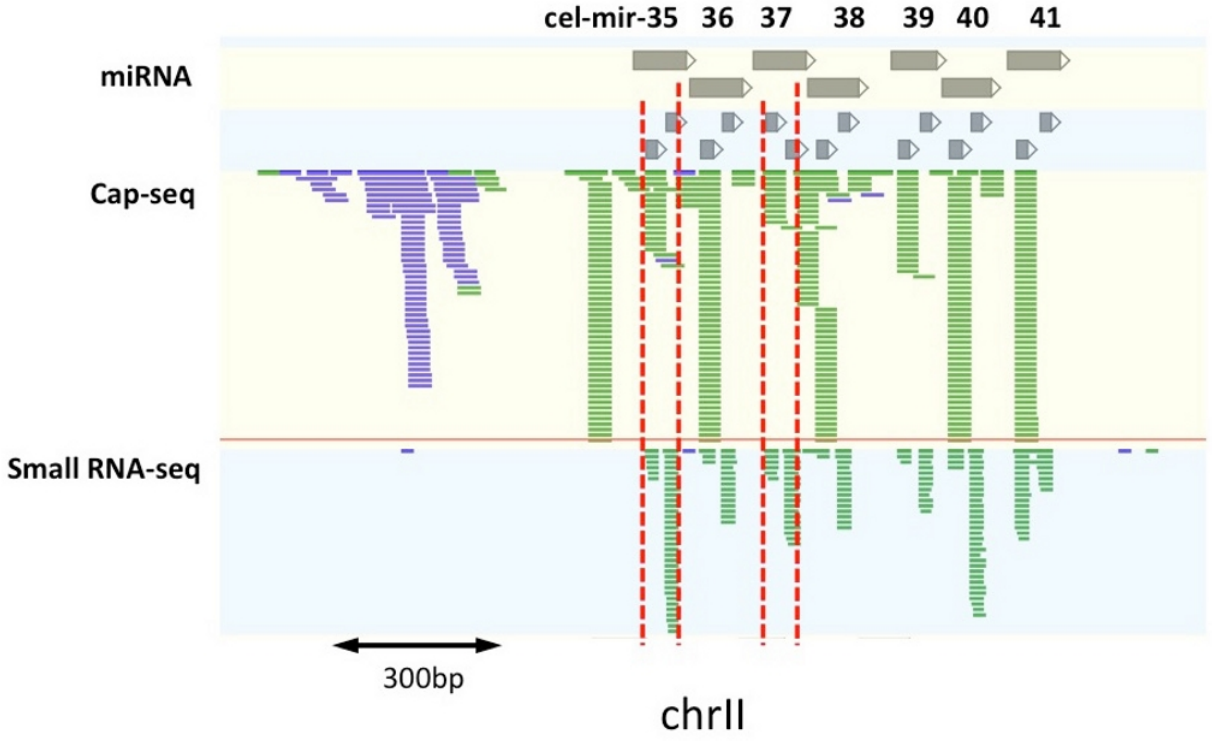

Figure 5.5: Capped TICs distribution in miRNA cluster cel-mir-35-41. Multiple strong capped peaks have been observed in pre-miRNAs in the cluster. As illustrated by the red dashed line, the Cap-seq peaks on the 5p arm have the same start position as the 5p-miRNAs, while the peaks on the 3p arm start from the end of the 3p-miRNAs. The length of Cap-seq reads is 36 nt.

5.3.5.1 5' re-cap after post-transcriptional processing

It has been reported that mature long transcripts of both protein-coding mRNAs and long ncRNAs in human cells can be processed post-transcriptionally to yield small RNAs, which are then

modified by the addition of a 5'-cap structure [44]. Later on, a cytoplasmic capping enzyme, which is able to add 5'-cap to the ends of cleaved RNAs was identified in murine erythroid and nonerythroid cells [110]. This cytoplasmic capping enzyme, together with a kinase, can transfer covalently bound GMP onto a 5'-monophosphate RNA to create a 5'-GpppX RNA, but it can not function on RNAs with 5'-hydroxyl ends [137]. This phenomenon is known as cytoplasmic capping, which has been found in both murine and human cells [110, 103, 64].

The well coordinated capped reads at 5p and 3p arms of pre-miRNA hairpins indicate that the exposed 5' ends of the miRNA transcripts after Drosha cleavage may be re-capped: that is why the capped reads were observed at the 5' ends of 5p miRNAs and the 3' ends of 3p miRNAs. Certainly, in the canonical miRNA biogenesis, pre-miRNAs are produced from pri-miRNAs in the nucleus by Drosha. Therefore, the 5' monophosphate ends at the cleavage sites may be re-capped by nuclear capping enzyme in the nucleus. However, pri-miRNAs in *C. elegans* are exported to the cytoplasm by XPO1 and be processed to pre-miRNAs either in the nuclear pore or in the cytoplasm [19]. Considering the discovery of cytoplasmic capping enzyme, the pre-miRNA re-capping is more likely to happen in the cytoplasm.

The processing of pri-miRNAs in the cytoplasm requires cytoplasmic miRNA processors. It has been shown that cytoplasmic RNA viruses which encode miRNAs were able to produce functional miRNAs in the cytoplasm of BHK-21 cells [125]. The processing of these virus-generated cytoplasmic pri-miRNAs also relies on Drosha but takes place in the cytoplasm [139]. Based on this discovery, although without direct experimental verification, there is a possibility that the similar cytoplasmic miRNA processors involving Drosha also exist in *C. elegans* cells. All these findings support a new model of miRNA biogenesis in *C. elegans* in which pri-miRNAs are exported to the cytoplasm by XPO1, where they are cleaved by Drosha and further processed. Transcripts of miRNA clusters may undergo cytoplasmic re-capping during the cleavage process (Figure 5.6).

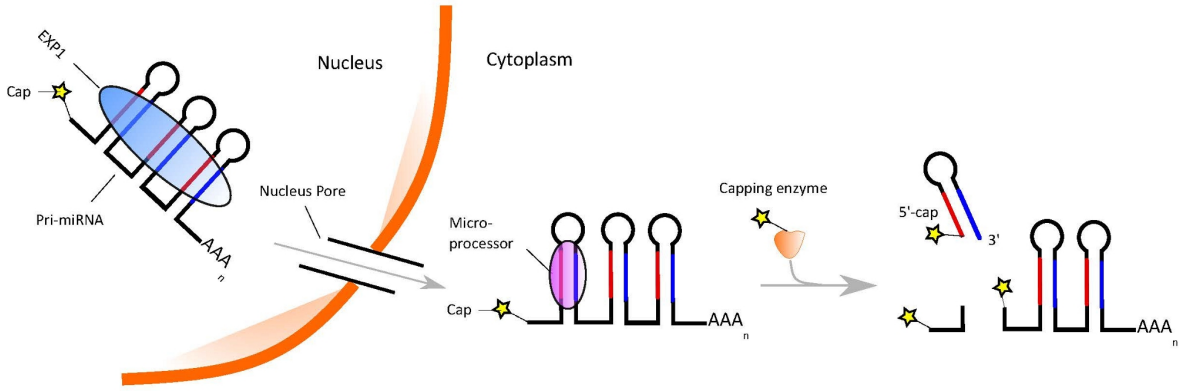

Figure 5.6: Model for miRNA cytoplasmic re-capping. In this non-canonical miRNA pathway, pri-miRNAs are exported to cytoplasm by XPO1 and processed there. During the pre-miRNA generating process, m^7G-caps are added to the 5' ends of newly generated pre-miRNA and pri-miRNA left over by cytoplasmic capping enzyme.

The cytoplasmic recapping of cluster miRNAs could produce capped 5p-miRNAs which may not be efficiently loaded on Ago. Indeed, from the annotations in miRBase, most of the clusters with capped TICs inside favorably generate 3p mature miRNAs. However, the clusters mir-41-44 and mir-86-8211 do prefer to generate 5p mature miRNAs. Here similar to the 5'-capped independent pre-miRNAs, the recapping may be a controlled process which only occur at certain cell types, developmental stages, or disease states. For example, with the capped short RNAs data from *C. elegans* young adult stage [49], we did not observe capped reads peaks inside of the miRNA clusters. The recapping could serve to regulate the strand selection on these miRNAs, suppressing the 5p-miRNAs and promoting the expression of their targets.

5.3.5.2 Multiple TSSs can be generated from the same promoter

Previous studies have shown that most core promoters do not have a single TSS, but multiple start sites that are closely located [135, 46]. The broad TSS promoters in *C. elegans* are often enriched for CpG island, while sharp TSS promoters often have TATA-box. For some promoters, although

TSSs are distributed over a large region, most transcription initiates at one specific nucleotide position [135]. Therefore, multiple capped peaks in individual pre-miRNAs or miRNA clusters may be generated from the broad promoter as multiple TSSs. To find evidence supporting this, we scanned the proximal promoters of these miRNA genes for motifs of TATA-box, Inr, DPE and BRE with position weight matrices derived from database JASPAR [134]. GC content and CpG number are also searched in the promoter regions (Table 5.1). We observe that the promoters of these miRNA clusters and individual 5'-capped miRNAs are usually GC rich: the GC content on each chromosome in *C. elegans* is almost the same, with a value of 36% $\pm$ 1%. But the GC content is usually above 40% for promoters of individual pre-miRNAs, and above 50% for miRNA clusters (Table 5.1). In addition, there tends to be a positive correlation between the GC content/CpG number and the number of capped peaks: higher GC content/more CpG result in more strong capped peaks.

Table 5.1: GC content, OE ratio and CpG number for miRNA promoters with multiple TSSs. (A) miRNA clusters. (B) individual miRNAs.

Promoters	Location	GC content	OE ratio	CPG number
(A) Cluster miRNAs	II:11537326-11537576	0.55	1.18	18
	II:11889425-11889675	0.56	1.09	13
	III:11937020-11937270	0.52	0.93	9
	III:2172175-2172425	0.55	1.0	15
	X:2368553-2368803	0.51	0.76	9
	X:13145204-13145454	0.6	1.34	24
(B) Individual miRNAs	cel-mir-244 I:4684364-4684614	0.4	0.82	8
	cel-mir-235 I:6162337-6162587	0.41	0.76	8
	cel-mir-238 III:8867375-8867625	0.4	0.66	6
	cel-mir-228 IV:5561825-5562075	0.43	1.18	12
	cel-mir-51 IV:11026062-11026312	0.51	1.09	17
	cel-mir-53 IV:11027641-11027891	0.45	1.38	16
	cel-mir-49 X:9989082-9989332	0.51	0.75	12

These capped reads peaks are correlated to the mature miRNA sequences (Figure 5.3, 5.5). There is a low probability that multiple peaks within miRNA cluster are simply generated randomly. It has been suggested that transcription start usage of each nucleotide can be predicted from local DNA sequence [46]. Therefore, it is likely that the positions of these TSSs are mainly determined by nucleotide sequence.

5.3.5.3 Pre-miRNAs in a cluster can be transcribed independently

Pre-miRNAs from the same genomic cluster can be transcribed and regulated independently [157]. For example, although the primary transcripts of mouse mir-433 and mir-127 were detected as overlapping in a 5'-3' unidirectional way, experiments have verified that they were transcribed independently from each other [143]. According to this model, the pre-miRNAs in the same cluster may transcribe independently, producing multiple 5' m^7G capped RNA peaks located inside of the cluster. The capped RNA peaks at the 5p arms indicate that the 5' end of the pre-miRNA may be determined by transcription initiation [161], while the capped peaks at the 3p arms may correspond to the TSSs for the subsequent pre-miRNAs. We noticed that many pre-miRNAs in the cluster have both the capped RNA peaks on its own 5' end and the 3p arm of upstream pre-miRNA. We have speculated that pre-miRNAs might be able to be generated in the canonical way with Drosha or in the non-canonical way by transcription initiation and termination, so here again, the same pre-miRNA may be generated by different mechanisms: one with Drosha and the other without.

5.3.6 Chromatin and Pol II profiles of primary miRNA promoters

5.3.6.1 Identification of divergent and multiple TSSs promoters

To characterize these identified pri-miRNA TSSs, we analysed the distribution of chromatin and Pol II features surrounding them. We observed that promoters in *C. elegans* often generate divergent or multiple transcripts with the Cap-seq datasets used. To identify these promoters, we defined the divergent/bidirectional and broad promoters (promoters with multiple TSSs) from those transcription initiation clusters (TICs). If the distance between two adjacent plus strand and minus strand TICs is less than or equal to 300 bp, they are combined as from the same divergent promoter. TICs on the same strand are clustered together if the distance between two adjacent TICs is within 500 bp. These clustered TICs on the same strand will define the broad promoters (see details in Materials and Methods). In the whole genome of *C. elegans*, we detected 11,272 promoters, of which 6,149 are divergent promoters and 2,359 are broad promoters (Figure 5.7(A)). The most upstream 5' ends of both plus and minus strand TICs were used to represent the TSSs of the promoter.

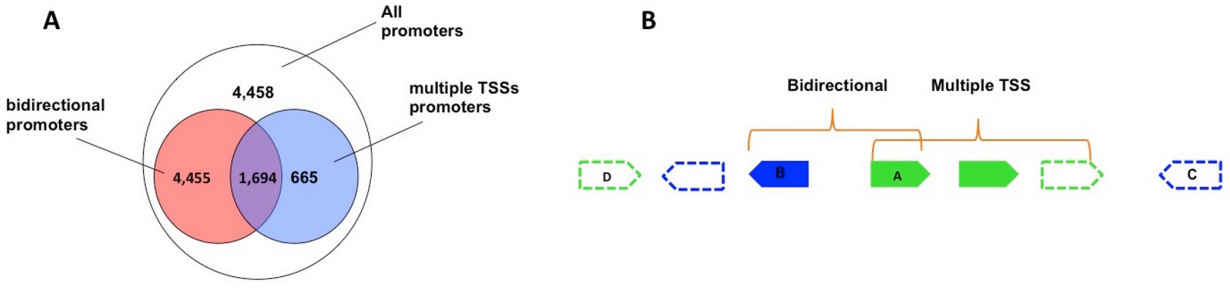

Figure 5.7: (A) Promoters distribution in *C. elegans*. (B) Detecting bidirectional and multiple transcription promoters in *C. elegans*.

With all the identified promoters, we focused on those with either multiple transcripts (at least 3 TSSs) or bidirectionality. That is, the broad promoters are non-bidirectional and bidirectional

promoters do not contain multiple TSSs. We aligned the TSSs of these promoters, and plotted H3K4me3 and Pol II signal profiles surrounding them (Figure 5.8 (A,B)). The result shows that H3K4me3 signal is much stronger in the downstream of broad promoters than bidirectional promoters, indicating that H3K4me3 is strongly correlated with transcription initiations. Interestingly, we observe that Pol II signal in the downstream of broad promoters is also slightly stronger than in the upstream, which would not be observed if we use all promoters with multiple TSSs (including bidirectional promoters (Figure S2 (D)). The higher level of downstream Pol II signal may be caused by the Pol II pausing at the transcription initiation sites. It has been shown that Pol II promoter-proximal pausing is rare in *C. elegans* [67], which may explain why the Pol II signal difference between upstream and downstream is not big.

5.3.6.2 Chromatin and Pol II profiles surrounding miRNA promoters

We then assigned these identified promoters to the intergenic miRNAs in *C. elegans*. For each of them, the flanking region between its 5' end and the most close upstream gene's 3' end is searched for promoters, and the most close promoter is assigned to the miRNA as its primary promoter. Again, we aligned these miRNAs with their primary TSSs acquired from the assigned promoters and plotted the H3K4me3 and Pol II signal surrouding the TSSs. The results are shown in Figure 5.8 (C,D). H3K4me3 signal surrounding miRNA broad promoters peaks at the identified TSSs while the signal surrounding bidirectional promoters peaks at the upstream 600 bp position, indicating that there may be much stronger minus transcripts for these bidirectional promoters. For Pol II signal, the intensities are similar in the upstream of both miRNA broad promoters and bidirectional promoters. However, again the downstream of broad promoters has a stronger Pol II signal, suggesting Pol II is also paused in the proximal promoters of miRNA genes.

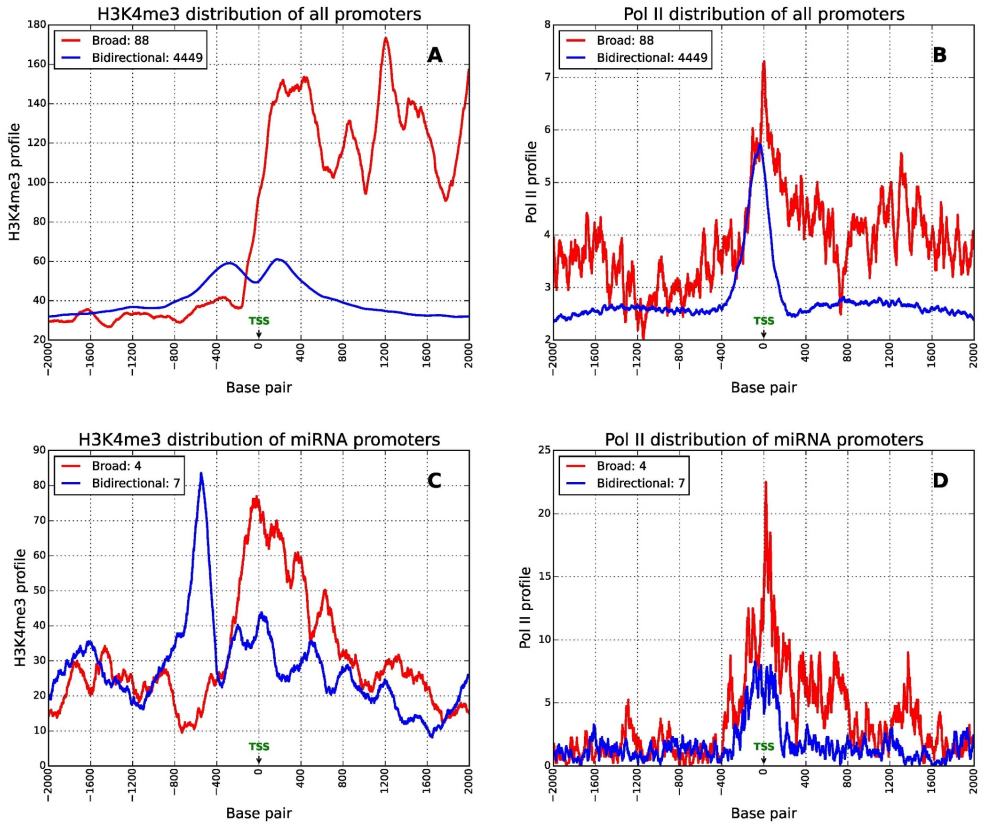

Figure 5.8: (A,B): H3K4me3 and Pol II signal profiles surrounding bidirectional and broad promoters. (C,D): H3K4me3 and Pol II signal profiles surrounding miRNA bidirectional and broad promoters.

5.4 Discussion

In this study, we used Cap-seq datasets to annotate the primary TSSs for intergenic miRNA genes to 1 base resolution in *C. elegans* and mouse. In total, we annotated the primary TSSs for 70 miRNAs and 8 miRNA clusters in *C. elegans*, and 37 miRNAs on mouse chromosome 7. Comparing with previous work using capped RNA-seq methods [49, 67], we have annotated the TSSs for many more miRNAs in both species. We noticed that the Cap-seq datasets we used are either generated from mixed-stage embryos (*C. elegans*) or mixed tissues (new born mouse), while the works we compared with only utilized few staged datasets in *C. elegans* [49, 67] or single tissue dataset in mouse [49]. Therefore, the limited tissue context may explain why fewer miRNA genes were

expressed and detected in their studies.

Similar to the previous study [161], which detected a special class of pre-miRNAs that are 5' m^7G capped in mice, here we also identified this type of pre-miRNAs in *C. elegans*. 5'-capped pre-miRNAs in mice prefer to be exported to the cytoplasm by XPO1 instead of XPO5 [161] . Comparing to mammals, XPO5 or its homologue is not encoded in *C. elegans* but XPO1 is [104]. Therefore, many more 5'-capped pre-miRNAs were expected to be identified in *C. elegans*. But using the Cap-seq datasets of mixed-stage embros of *C. elegans*, we have only detected 9 candidate 5'-capped pre-miRNAs. The XPO1 and cap-binding complex (CBC) have been reported to act jointly to export pri-miRNAs in *C. elegans* and Drosophila, but how the subsequent pre-miRNAs are processed from these pri-miRNAs in the cytoplasm remains unclear [19]. Accordingly, both 5'-capped pre-miRNAs and normal pri-miRNAs can be exported to the cytoplasm and their subsequent processing may be different from that in the canonical miRNA biogenesis pathway. Since XPO1 does not specifically export 5'-capped pre-miRNAs in *C. elegans*, its existence does not necessarily result in more 5'-capped pre-miRNAs. Thus, it is not odd that only a small number of these unusual pre-miRNAs are observed in *C. elegans*.

It has been suggested that XPO1-dependent m^7G capped pre-miRNAs may represent a group of ancient miRNAs that appeared before the emergence of XPO5 [161]. Therefore, these m^7G capped pre-miRNAs may be well conserved in different lineages. We checked the conservation of those identified 5'-capped pre-miRNAs in *C. elegans* and mouse from miRviewer [62]. Surprisingly, almost all of these identified m^7G capped pre-miRNAs are lineage specific: they are detected either only in caenorhhabditis or muroidea.

In both mouse and *C. elegans*, we have identified a class of pre-miRNAs that are 5' m^7G capped but also posses upstream candidate primary TSSs. It has been suggested that most genes in mammals are not transcribed from a single TSS, but multiple TSSs that are closely located

over 50 to 100 nucleotides [21, 135, 46, 36]. Indeed, we observed that some upstream TICs are very close to the pre-miRNAs, suggesting that the proximal core promoter of these miRNA genes may generate multiple transcription initiations that are closely located to each other. However, the other further upstream TICs (over 500 nt) are unlikely to be generated from the same promoter as the pre-cap TICs. Many genes have alternative promoters and can generate multiple isoforms in different cell types, tissues or developmental stages [69, 7]. Since we used the Cap-seq datasets of mixed staged embryos of *C. elegans*, these miRNA genes may utilize alternative promoters at different stages to generate diverse transcripts which are subject to distinct processing procedures. That is, these 5'-capped pre-miRNAs may be only generated in specific cell types, developmental stages or disease states. Considering that these XPO1-dependent 5' m^7G capped pre-miRNAs may belong to a group of ancient miRNAs, it is reasonable that at least some of them should be conserved along different lineages during evolution. However, we have shown that almost all of these identified 5' m^7G capped pre-miRNAs are lineage specific. There is a possibility that these miRNA genes are newly generated and acquire their ability of producing 5'-capped pre-miRNAs later in the evolution. In extreme cases, most of these 5'-capped pre-miRNAs may have upstream distal promoters that can produce the normal primary transcripts at the other time. The situation where m^7G capped pre-miRNAs are expressed may due to the lack of Drosha or other related microprocessor. The ability of generating the same miRNAs with or without Drosha may help the organism to better adapt to the complex and changeable environment.

Many of the identified 5' m^7G capped pre-miRNAs in *C. elegans* preferentially generate 5p-miRNAs, which is different from that in mice as previously reported. These pre-miRNAs usually also have upstream candidate primary TSSs. We surmised that these 5'-capped pre-miRNAs may produce the 5p-miRNAs using the alternative upstream primary TSSs. However, this still needs further efforts to clarify.

Multiple capped peaks have been observed within pre-miRNAs in a cluster, and the well co-ordinated relationship between the capped peaks and the mature miRNAs suggest that the 5' m^7G cap may be added during the pre-miRNA generating process. We proposed a new model of miRNA biogenesis in which the pri-miRNAs are cleaved and capped in the cytoplasm simultaneously (Figure 5.6). We noticed that the cytoplasmic capping is prominent in miRNA clusters, likely because the exposed 5' end of miRNA cluster transcripts during cleavage need to be protected. Therefore, the m^7G cap is added and protects the cleaved transcripts from being degraded during the cleavage process. Then, the subsequent pre-miRNAs can be generated successfully.

The pre-miRNAs in clusters may be transcribed independently. To gain more evidence for this model, we also looked at the the sequence motif surrounding the modes of pre-cap TICs within clustered pre-miRNAs. The YR motif, which was shown at the putative primary miRNA TSSs and pre-cap TSSs of independent pre-miRNAs, is not observed (Figure S8). Instead, a weak consensus sequence of "TNGG" is detected, in which "N" locates at the +1 position representing the modes of the TICs. Therefore, at least for some miRNA clusters, the independent transcription model is not supported by the YR motif.

Chapter 6

Conclusion and Future work

6.1 Conlusions

Characterizing the composition and functions of microbial communities from host-related or environmental samples using metagenomic sequencing has become a favorable method. In this study, we focused on characterizing the virus quasispecies from viral metagenomic data. Towards this goal, we developed a pipeline consisting of three tools for assembling viral strains in a quasispecies and estimating haplotype number and corresponding abundances. The three tools are TAR-VIR, PEHaplo, and VirBin.

TAR-VIR is the tool for enriching reads of targeted viruses with remote homolog or partial references from metagenomic data. Compared to the standard method that relies on read mapping to classify reads of different viruses, TAR-VIR has higher sensitivity to identify divergent strains of known virus families, which is a key function for fast-mutating viruses such as HIV, HCV etc. It applied an external overlap extension step to iteratively recruit reads from initially aligned "seed" reads. The overlap detection between large-scale metagenomic sequencing reads was implemented in the efficient Burrows-Wheeler Transform (BWT) and FM-index algorithms. TAR-VIR can be used as a preliminary step to isolate and enrich reads from metagenomic data before assembly even without high-quality references.

PEHaplo is the *de novo* assembly tools for reconstructing viral haplotypes from virus quasispecies sequencing data. It takes advantage of the long insert size of paired-end sequencing to

differentiate haplotypes sharing long common regions. The methods of PEHaplo are based on overlap graph, which reconstruct haplotypes by finding the paths well-supported by paired-end reads. While paired-end information has been used by many other assembly tools, PEHaplo did a thorough analysis of the relationship between LCS and the insert size, and dived deeply into the utilization of paired-end reads for removing false edges and finding correct paths. The benchmark results with several recently published viral quasispecies assembly tools demonstrate that PEHaplo is able to produce accurate and much longer contigs.

The output of PEHaplo are contigs, which are partial sequences for viral genomes. To better characterize a viral quasispecies, we need to know the number of viral haplotypes and their corresponding abundances. VirBin is the binning tool that takes assembled contigs as input, estimates the number of haplotypes, clusters contigs into groups, and calculates the abundances for each haplotype. The binning problem for viral quasispecies is very different to the canonical species-level binning problems because commonly adopted features such as k-mer frequency are not able to distinguish different strains. VirBin takes advantage of the high sequence similarities between viral strains in a quasispecies, estimates the number of haplotypes by aligning contigs to each other. The relative abundances for each contig can also be more accurately calculated from the contig alignment results. Based on the contig abundances, VirBin applies an EM algorithm to cluster contigs into different groups. The benchmark results with tool MaxBin [160] have shown its superiority on both simulated or real viral quasispecies data.

At the beginning of my Ph.D. study, I also worked on a project for studying the transcriptional regulation of miRNA genes in C. elegans and mice. In this project, we utilized Cap-seq data to identify the primary transcriptional start sites for miRNA genes, characterized a special group of miRNAs whose precursors are 5'-capped and may be directly generated by transcription.

6.2 Future work

6.2.1 Reference-free viral sequences classification

While TAR-VIR is able to recruit viral reads with partial or remotely related homologous references, the method is not applicable to new viruses without any available references. Therefore, directly classifying viral reads from metagenomic data without references may still be needed. For directly classifying viral reads, a fundamental question to ask is whether viral sequences can be differentiated from other bacterial or eukaryotic sequences. However, the canonical sequence analysis (such as k-mer frequencies) results on NGS reads showed that viral reads are highly diverse and are difficult to be differentiated from other sequences.

The deep learning methods, with multiple layers of artificial neural networks and numerous trainable parameters, are powerful for classification. They do not require elaborately designed sequence features as input and can take simply encoded raw sequences as input for classification. Whether we can take advantage of deep learning models for directly classifying viral reads from metagenomic data sets is worth exploring.

6.2.2 Virus quasispecies assembly future work

There are still some computational challenges remaining in virus quasispecies assembly.

Assembly of low abundance haplotypes One of the challenges is that rare haplotypes are difficult to assemble from the data. When the sequencing coverage is low for a viral strain, there may not be enough overlaps between reads, resulting in a disjoined overlap graph with multiple small connected subgraphs. In addition, even if sequencing depths for rare haplotypes are deep enough, the edges between shared nodes and rare haplotype nodes may be suppressed by the edges between shared nodes and abundant haplotype nodes, resulting in fragmented contigs.

LCS is longer than paired-end insert size Another challenge is the situation when LCS between two haplotypes is much longer than the average paired-end insert size. In this case, few read pairs can go across the common nodes, thus finding correct paths from the graph is difficult. The only available feature is the reads coverage. When the coverages for these two haplotypes are significantly difficult, the algorithm is able to choose the successor with similar coverage to the current path for appending. However, if the sequencing coverages of the two haplotypes are similar, the path extending is hard. With short reads from NGS, one of the possible solutions is to increase the insert size. Alternatively, we may take advantage of the long reads from third-generation sequencing.

6.2.3 Binning low-quality contigs

To correctly estimate the number of haplotypes and calculate the relative abundances for contigs, the contigs need to be aligned, and windows identification heavily depends on the alignment. However, when there are chimeric contigs or contigs with errors, either indels/mismatches, the alignment may not be accurate and eventually result in an inaccurate estimation of haplotype number or clustering. How to develop/improve the binning algorithms to robustly handle low-quality contigs is another problem to be further investigated.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] T. Abeel, T. Van Parys, Y. Saeys, J. Galagan, and Y. Van de Peer. GenomeView: a next-generation genome browser. *Nucl. Acids Res.*, 40(2):e12, 2012.

[2] A. Allam, P. Kalnis, and V. Solovyev. Karect: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data. *Bioinformatics*, page btv415, 2015.

[3] R. Andino and E. Domingo. Viral quasispecies. *Virology*, 479:46–51, 2015.

[4] I. Astrovskaya, B. Tork, S. Mangul, K. Westbrooks, I. Măndoiu, P. Balfe, and A. Zelikovsky. Inferring viral quasispecies spectra from 454 pyrosequencing reads. *BMC bioinformatics*, 12(6):S1, 2011.

[5] J. Baaijens, A. Z. El Aabidine, E. Rivals, and A. Schoenhuth. De novo assembly of viral quasispecies using overlap graphs. *bioRxiv*, page 080341, 2016.

[6] J. A. Baaijens, A. Z. El Aabidine, E. Rivals, and A. Schönhuth. De novo assembly of viral quasispecies using overlap graphs. *Genome research*, 27(5):835–848, 2017.

[7] D. Baek, C. Davis, B. Ewing, D. Gordon, and P. Green. Characterization and predictive discovery of evolutionarily conserved mammalian alternative promoters. *Genome Res.*, 17(2):145–155, 2007.

[8] A. Bankevich, S. Nurk, D. Antipov, A. A. Gurevich, M. Dvorkin, A. S. Kulikov, V. M. Lesin, S. I. Nikolenko, S. Pham, A. D. Prjibelski, et al. SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *Journal of computational biology*, 19(5):455–477, 2012.

[9] V. C. Barbosa, R. Donangelo, and S. R. Souza. Quasispecies dynamics with network constraints. *Journal of theoretical biology*, 312:114–119, 2012.

[10] E. Berezikov. Evolution of microRNA diversity and regulation in animals. *Nature Rev. Genet.*, 12(12):846–860, Dec. 2011.

[11] E. Berezikov, W.-J. Chung, J. Willis, E. Cuppen, and E. C. Lai. Mammalian mirtron genes. *Mol. Cell*, 28(2):328–336, 2007.

[12] M. Biasiolo, G. Sales, M. Lionetti, L. Agnelli, K. Todoerti, A. Bisognin, A. Coppe, C. Romualdi, A. Neri, and S. Bortoluzzi. Impact of host genes and strand selection on miRNA and miRNA* expression. *PLoS ONE*, 6(8):1–11, 2011.

[13] C. K. Biebricher and M. Eigen. The error threshold. *Virus research*, 107(2):117–127, 2005.

[14] S. Boisvert, F. Raymond, É. Godzaridis, F. Laviolette, and J. Corbeil. Ray meta: scalable de novo metagenome assembly and profiling. *Genome biology*, 13(12):R122, 2012.

[15] A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014.

[16] G. M. Borchert, W. Lanier, and B. L. Davidson. RNA polymerase III transcribes human microRNAs. *Nature Struct. Mol. Biol.*, 13(12):1097–1101, 2006.

[17] J. Bracht, S. Hunter, R. Eachus, P. Weeks, and A. E. Pasquinelli. Trans-splicing and polyadenylation of let-7 microRNA primary transcripts. *RNA*, 10(10):1586–1594, 2004.

[18] M. Burrows and D. J. Wheeler. A block-sorting lossless data compression algorithm. 1994.

[19] I. Büssing, S. Yang Jr, E. C. Lai, and H. Großhans. The nuclear export receptor XPO-1 supports primary miRNA processing in C. elegans and Drosophila. *EMBO J.*, 29(11):1830–1839, 2010.

[20] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden. BLAST+: architecture and applications. *BMC bioinformatics*, 10(1):421, 2009.

[21] P. Carninci, A. Sandelin, B. Lenhard, S. Katayama, K. Shimokawa, J. Ponjavic, C. A. Semple, M. S. Taylor, P. G. Engström, M. C. Frith, et al. Genome-wide analysis of mammalian promoter architecture and evolution. *Nature Genet.*, 38(6):626–635, 2006.

[22] M. J. Chaisson, D. Brinza, and P. A. Pevzner. De novo fragment assembly with short mate-paired reads: Does the read length matter? *Genome research*, 19(2):336–346, 2009.

[23] M. CHAN-YEUNG and R.-H. XU. SARS: epidemiology. *Respirology*, 8(s1), 2003.

[24] T.-C. Chang, M. Pertea, S. Lee, S. L. Salzberg, and J. T. Mendell. Genome-wide annotation of microrna primary transcript structures reveals novel regulatory mechanisms. *Genome research*, 25(9):1401–1409, 2015.

[25] J. Chen, Z. Dai, C. Cao, Q. Zhang, H. Liu, and X. Sun. Next-generation sequencing data processing: Analysis of unmapped reads and extremely high mapped peaks. In *Biomedical Engineering and Informatics (BMEI), 2012 5th International Conference on*, pages 893–897. IEEE, 2012.

[26] J. Chen, Y. Zhao, and Y. Sun. De novo haplotype reconstruction in viral quasispecies using paired-end read guided path finding. *Bioinformatics*, 1:9, 2018.

[27] R. A.-J. Chen, T. A. Down, P. Stempor, Q. B. Chen, T. A. Egelhofer, L. W. Hillier, T. E. Jeffers, and J. Ahringer. The landscape of RNA polymerase II transcription initiation in C. elegans reveals promoter and enhancer architectures. *Genome Res.*, 23(8):1339–1347,

2013.

[28] T. P. Chendrimada, R. I. Gregory, E. Kumaraswamy, J. Norman, N. Cooch, K. Nishikura, and R. Shiekhattar. TRBP recruits the Dicer complex to Ago2 for microRNA processing and gene silencing. *Nature*, 436(7051):740–744, 2005.

[29] C.-H. Chien, Y.-M. Sun, W.-C. Chang, P.-Y. Chiang-Hsieh, T.-Y. Lee, W.-C. Tsai, J.-T. Horng, A.-P. Tsou, and H.-D. Huang. Identifying transcriptional start sites of human microRNAs based on high-throughput sequencing data. *Nucl. Acids Res.*, 39(21):9345–9356, 2011.

[30] F. S. Collins and H. Varmus. A new initiative on precision medicine. *New England Journal of Medicine*, 372(9):793–795, 2015.

[31] D. L. Corcoran, K. V. Pandit, B. Gordon, A. Bhattacharjee, N. Kaminski, and P. V. Benos. Features of mammalian microRNA promoters emerge from polymerase II chromatin immunoprecipitation data. *PLoS ONE*, 4(4):e5279, 2009.

[32] C. Coronnello and P. V. Benos. Comir: combinatorial microrna target prediction tool. *Nucl. Acids Res.*, 41(W1):W159–W164, 2013.

[33] A. K. Corsi. A biochemist's guide to Caenorhabditis elegans. *Anal. Biochem.*, 359(1):1–17, 2006.

[34] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner. WebLogo: a sequence logo generator. *Genome Res.*, 14(6):1188–1190, 2004.

[35] L. Dai, K. Chen, B. Youngren, J. Kulina, A. Yang, Z. Guo, J. Li, P. Yu, and S. Gu. Cytoplasmic drosha activity generated by alternative splicing. *Nucleic Acids Research*, page gkw668, 2016.

[36] Y. M. Danino, D. Even, D. Ideses, and T. Juven-Gershon. The core promoter: at the heart of gene expression. *Biochim. Biophys. Acta*, 1849(8):1116–1131, 2015.

[37] R. V. Davuluri, Y. Suzuki, S. Sugano, C. Plass, and T. H.-M. Huang. The functional consequences of alternative promoter use in mammalian genomes. *Trends Genet.*, 24(4):167–177, 2008.

[38] M. de Hoon and Y. Hayashizaki. Deep cap analysis gene expression (CAGE): genome-wide identification of promoters, quantification of their expression, and network inference. *Biotechniques*, 44(5):627, 2008.

[39] F. Di Giallonardo, A. Töpfer, M. Rey, S. Prabhakaran, Y. Duport, C. Leemann, S. Schmutz, N. K. Campbell, B. Joos, M. R. Lecca, et al. Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. *Nucleic acids research*, 42(14):e115–e115,

2014.

[40] S. Djebali, C. A. Davis, A. Merkel, A. Dobin, T. Lassmann, A. Mortazavi, A. Tanzer, J. Lagarde, W. Lin, F. Schlesinger, et al. Landscape of transcription in human cells. *Nature*, 489(7414):101–108, 2012.

[41] E. Domingo, J. Sheldon, and C. Perales. Viral quasispecies evolution. *Microbiology and Molecular Biology Reviews*, 76(2):159–216, 2012.

[42] M. Eigen. Error catastrophe and antiviral strategy. *Proceedings of the National Academy of Sciences*, 99(21):13374–13376, 2002.

[43] M. Fedurco, A. Romieu, S. Williams, I. Lawrence, and G. Turcatti. Bta, a novel reagent for dna attachment on glass and efficient generation of solid-phase amplified dna colonies. *Nucleic acids research*, 34(3):e22–e22, 2006.

[44] K. Fejes-Toth, V. Sotirova, R. Sachidanandam, G. Assaf, G. J. Hannon, P. Kapranov, S. Foissac, A. T. Willingham, R. Duttagupta, E. Dumais, et al. Post-transcriptional processing generates a diversity of 5'-modified long and short RNAs. *Nature*, 457(7232):1028–1032, 2009.

[45] K. A. Frazer, L. Pachter, A. Poliakov, E. M. Rubin, and I. Dubchak. VISTA: computational tools for comparative genomics. *Nucleic acids research*, 32(suppl_2):W273–W279, 2004.

[46] M. C. Frith, E. Valen, A. Krogh, Y. Hayashizaki, P. Carninci, and A. Sandelin. A code for transcription initiation in mammalian genomes. *Genome Res.*, 18(1):1–12, 2008.

[47] G. Gonnella and S. Kurtz. Readjoiner: a fast and memory efficient string graph-based sequence assembler. *BMC bioinformatics*, 13(1):82, 2012.

[48] S. Griffiths-Jones, R. J. Grocock, S. Van Dongen, A. Bateman, and A. J. Enright. miRBase: microRNA sequences, targets and gene nomenclature. *Nucl. Acids Res.*, 34(suppl 1):D140–D144, 2006.

[49] W. Gu, H.-C. Lee, D. Chaves, E. M. Youngman, G. J. Pazour, D. Conte Jr, and C. C. Mello. CapSeq and CIP-TAP Identify Pol II Start Sites and Reveal Capped Small RNAs as C. elegans piRNA Precursors. *Cell*, 151(7):1488–1500, 2012.

[50] D. Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge university press, 1997.

[51] M. Haj Rachid and Q. Malluhi. A practical and scalable tool to find overlaps between sequences. *BioMed research international*, 2015, 2015.

[52] J. J. Holland, E. Domingo, J. C. de la Torre, and D. A. Steinhauer. Mutation frequencies at

defined single codon sites in vesicular stomatitis virus and poliovirus can be increased only slightly by chemical mutagenesis. *Journal of virology*, 64(8):3960, 1990.

[53] L. Z. Hong, S. Hong, H. T. Wong, P. P. Aw, Y. Cheng, A. Wilm, P. F. de Sessions, S. G. Lim, N. Nagarajan, M. L. Hibberd, et al. Base-seq: a method for obtaining long viral haplotypes from short sequence reads. *Genome biology*, 15(11):517, 2014.

[54] A. Huang, R. Kantor, A. DeLong, L. Schreier, and S. Istrail. Qcolors: An algorithm for conservative viral quasispecies reconstruction from short and non-contiguous next generation sequencing reads. *In silico biology*, 11(5, 6):193–201, 2011.

[55] W. Huang, L. Li, J. R. Myers, and G. T. Marth. ART: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2011.

[56] W. Huang, L. Li, J. R. Myers, and G. T. Marth. Art: a next-generation sequencing read simulator. *Bioinformatics*, 28(4):593–594, 2012.

[57] M. Hunt, A. Gall, S. H. Ong, J. Brener, B. Ferns, P. Goulder, E. Nastouli, J. A. Keane, P. Kellam, and T. D. Otto. Iva: accurate de novo assembly of rna virus genomes. *Bioinformatics*, 31(14):2374–2376, 2015.

[58] J. L. Jameson and D. L. Longo. Precision medicine—personalized, problematic, and promising. *Obstetrical & Gynecological Survey*, 70(10):612–614, 2015.

[59] D. Jayasundara, I. Saeed, S. Maheswararajah, B. Chang, S.-L. Tang, and S. K. Halgamuge. ViQuaS: an improved reconstruction pipeline for viral quasispecies spectra generated by next-generation sequencing. *Bioinformatics*, 31(6):886–896, 2014.

[60] J. Kärkkäinen and P. Sanders. Simple linear work suffix array construction. In *International Colloquium on Automata, Languages, and Programming*, pages 943–955. Springer, 2003.

[61] T. Kasai, G. Lee, H. Arimura, S. Arikawa, and K. Park. Linear-time longest-common-prefix computation in suffix arrays and its applications. In *Annual Symposium on Combinatorial Pattern Matching*, pages 181–192. Springer, 2001.

[62] A. Kiezun, S. Artzi, S. Modai, N. Volk, O. Isakov, and N. Shomron. miRviewer: a multispecies microRNA homologous viewer. *BMC Res. Notes*, 5(1):1–6, 2012.

[63] V. N. Kim and J.-W. Nam. Genomics of microRNA. *Trends Genet.*, 22(3):165–173, Mar. 2006.

[64] D. L. Kiss, K. Oman, R. Bundschuh, and D. R. Schoenberg. Uncapped 5' ends of mRNAs targeted by cytoplasmic capping map to the vicinity of downstream CAGE tags. *FEBS Lett.*, 589(3):279–284, 2015.

[65] R. Kodzius, M. Kojima, H. Nishiyori, M. Nakamura, S. Fukuda, M. Tagami, D. Sasaki, K. Imamura, C. Kai, M. Harbers, et al. CAGE: cap analysis of gene expression. *Nat. Methods*, 3(3):211–222, 2006.

[66] J. Krol, I. Loedige, and W. Filipowicz. The widespread regulation of microRNA biogenesis, function and decay. *Nature Rev. Genet.*, 11(9):597–610, Sept. 2010.

[67] W. S. Kruesi, L. J. Core, C. T. Waters, J. T. Lis, and B. J. Meyer. Condensin controls recruitment of RNA polymerase II to achieve nematode X-chromosome dosage compensation. *Elife*, 2:e00808, 2013.

[68] A. Kuehbacher, C. Urbich, A. M. Zeiher, and S. Dimmeler. Role of Dicer and Drosha for endothelial microRNA expression and angiogenesis. *Circ. Res.*, 101(1):59–68, 2007.

[69] J.-R. Landry, D. L. Mager, and B. T. Wilhelm. Complex controls: the role of alternative promoters in mammalian genomes. *Trends Genet.*, 19(11):640–648, 2003.

[70] B. Langmead and S. L. Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357–359, 2012.

[71] B. Langmead and S. L. Salzberg. Fast gapped-read alignment with bowtie 2. *Nature methods*, 9(4):357–359, 2012.

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

[73] J. Laserson, V. Jojic, and D. Koller. Genovo: de novo assembly for metagenomes. *J Comput Biol*, 18(3):429–443, 2011.

[74] N. C. Lau, L. P. Lim, E. G. Weinstein, and D. P. Bartel. An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans. *Science*, 294(5543):858–862, 2001.

[75] A. S. Lauring and R. Andino. Quasispecies theory and the behavior of rna viruses. *PLoS Pathog*, 6(7):e1001005, 2010.

[76] Y. Lee, C. Ahn, J. Han, H. Choi, J. Kim, J. Yim, J. Lee, P. Provost, O. Rådmark, S. Kim, et al. The nuclear RNase III Drosha initiates microRNA processing. *Nature*, 425(6956):415–419, 2003.

[77] Y. Lee, M. Kim, J. Han, K.-H. Yeom, S. Lee, S. H. Baek, and V. N. Kim. MicroRNA genes are transcribed by RNA polymerase II. *EMBO J.*, 23(20):4051–4060, 2004.

[78] J. Lei and Y. Sun. miR-PREFeR: an accurate, fast and easy-to-use plant miRNA prediction tool using small RNA-Seq data. *Bioinformatics*, 30(19):2837–2839, 2014.

[79] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows–Wheeler transform. *bioinformatics*, 25(14):1754–1760, 2009.

[80] L. Li, X. Deng, A. C. Da Costa, R. Bruhn, S. G. Deeks, and E. Delwart. Virome analysis of antiretroviral-treated HIV patients shows no correlation between T-cell activation and anelloviruses levels. *Journal of Clinical Virology*, 72:106–113, 2015.

[81] M. Li, B. Wu, X. Yan, J. Luo, Y. Pan, F.-X. Wu, and J. Wang. Pecc: correcting contigs based on paired-end read distribution. *Computational biology and chemistry*, 69:178–184, 2017.

[82] N. Li, X. You, T. Chen, S. D. Mackowiak, M. R. Friedländer, M. Weigt, H. Du, A. Gogol-Döring, Z. Chang, C. Dieterich, et al. Global profiling of miRNAs and the hairpin precursors: insights into miRNA processing and novel miRNA discovery. *Nucl. Acids Res.*, 41(6):3619–3634, 2013.

[83] R. Li, H. Zhu, J. Ruan, W. Qian, X. Fang, Z. Shi, Y. Li, S. Li, G. Shan, K. Kristiansen, et al. De novo assembly of human genomes with massively parallel short read sequencing. *Genome research*, 20(2):265–272, 2010.

[84] Y. Li, H. Wang, K. Nie, C. Zhang, Y. Zhang, J. Wang, P. Niu, and X. Ma. VIP: an integrated pipeline for metagenomics of virus identification and discovery. *Scientific reports*, 6, 2016.

[85] E. S. Lim, Y. Zhou, G. Zhao, I. K. Bauer, L. Droit, I. M. Ndao, B. B. Warner, P. I. Tarr, D. Wang, and L. R. Holtz. Early life dynamics of the human gut virome and bacterial microbiome in infants. *Nature medicine*, 21(10):1228–1234, 2015.

[86] H.-H. Lin and Y.-C. Liao. drVM: a new tool for efficient genome assembly of known eukaryotic viruses from metagenomes. *GigaScience*, 6(2):1–10, 2017.

[87] C.-C. Lo and P. S. Chain. Rapid evaluation and quality control of next generation sequencing data with FaQCs. *BMC bioinformatics*, 15(1):366, 2014.

[88] Y. Y. Lu, T. Chen, J. A. Fuhrman, and F. Sun. Cocacola: binning metagenomic contigs using sequence composition, read coverage, co-alignment and paired-end read linkage. *Bioinformatics*, 33(6):791–798, 2017.

[89] C. Luo, R. Knight, H. Siljander, M. Knip, R. J. Xavier, and D. Gevers. Constrains identifies microbial strains in metagenomic datasets. *Nature biotechnology*, 33(10):1045, 2015.

[90] C. Luo, D. Tsementzi, N. Kyrpides, and K. Konstantinidis. Individual genome assembly from complex community short-read metagenomic datasets. *ISME J.*, 6(4):898–901, 2012.

[91] T. F. Lüscher. Frontiers in precision medicine: genes and their modulation by mirnas. *European Heart Journal*, 37(43):3247–3250, 2016.

[92] R. Malhotra, S. Prabhakara, M. Poss, and R. Acharya. Estimating viral haplotypes in a population using k-mer counting. In *IAPR International Conference on Pattern Recognition in Bioinformatics*, pages 265–276. Springer, 2013.

[93] R. Malhotra, M. M. S. Wu, A. Rodrigo, M. Poss, and R. Acharya. Maximum likelihood de novo reconstruction of viral populations using paired end sequencing data. *arXiv preprint arXiv:1502.04239*, 2015.

[94] S. K. Mallanna and A. Rizzino. Emerging roles of microRNAs in the control of embryonic stem cells and the generation of induced pluripotent stem cells. *Dev. Biol.*, 344(1):16–25, 2010.

[95] S. Mangul, N. C. Wu, N. Mancuso, A. Zelikovsky, R. Sun, and E. Eskin. VGA: a method for viral quasispecies assembly from ultra-deep sequencing data. In *Computational Advances in Bio and Medical Sciences (ICCABS), 2014 IEEE 4th International Conference on*, pages 1–1. IEEE, 2014.

[96] F. Matin, V. Jeet, J. A. Clements, G. M. Yousef, and J. Batra. Microrna theranostics in prostate cancer precision medicine. *Clinical Chemistry*, pages clinchem–2015, 2016.

[97] H. A. Meijer, E. M. Smith, and M. Bushell. Regulation of miRNA strand selection: follow the leader? *Biochem. Soc. Trans.*, 42(4):1135–1140, 2014.

[98] M. L. Metzker. Sequencing technologies—the next generation. *Nature reviews genetics*, 11(1):31–46, 2010.

[99] F. Meyer, D. Paarmann, M. D'Souza, R. Olson, E. M. Glass, M. Kubal, T. Paczian, A. Rodriguez, R. Stevens, A. Wilke, et al. The metagenomics RAST server–a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC bioinformatics*, 9(1):386, 2008.

[100] A. Mikheenko, V. Saveliev, and A. Gurevich. Metaquast: evaluation of metagenome assemblies. *Bioinformatics*, page btv697, 2015.

[101] R. D. Mitra and G. M. Church. In situ localized amplification and contact replication of many individual dna molecules. *Nucleic Acids Research*, 27(24):e34–e39, 1999.

[102] C. M. Mizuno, F. Rodriguez-Valera, N. E. Kimes, and R. Ghai. Expanding the marine virosphere using metagenomics. *PLoS genetics*, 9(12):e1003987, 2013.

[103] C. Mukherjee, D. P. Patil, B. A. Kennedy, B. Bakthavachalu, R. Bundschuh, and D. R. Schoenberg. Identification of cytoplasmic capping targets reveals a role for cap homeostasis in translation and mRNA stability. *Cell Rep.*, 2(3):674–684, 2012.

[104] D. Murphy, B. Dancis, and J. R. Brown. The evolution of core proteins involved in mi-

croRNA biogenesis. *BMC Evol. Biol.*, 8(1):1–18, 2008.

[105] S. N. Naccache, S. Federman, N. Veeraraghavan, M. Zaharia, D. Lee, E. Samayoa, J. Bouquet, A. L. Greninger, K.-C. Luk, B. Enge, et al. A cloud-compatible bioinformatics pipeline for ultrarapid pathogen identification from next-generation sequencing of clinical samples. *Genome research*, 24(7):1180–1192, 2014.

[106] T. Namiki, T. Hachiya, H. Tanaka, and Y. Sakakibara. Metavelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Research*, 40(20):e155, 2012.

[107] M. A. Nowak. *Evolutionary dynamics*. Harvard University Press, Cambridge Massachusetts, 2006.

[108] S. Nurk, D. Meleshko, A. Korobeynikov, and P. A. Pevzner. metaSPAdes: a new versatile metagenomic assembler. *Genome research*, pages gr–213959, 2017.

[109] N. D. Olson, T. J. Treangen, C. M. Hill, V. Cepeda-Espinoza, J. Ghurye, S. Koren, and M. Pop. Metagenomic assembly through the lens of validation: recent advances in assessing and improving the quality of genomes assembled from metagenomes. *Briefings in bioinformatics*, 2017.

[110] Y. Otsuka, N. L. Kedersha, and D. R. Schoenberg. Identification of a cytoplasmic complex that adds a cap onto 5'-monophosphate RNA. *Mol. Cell Biol.*, 29(8):2155–2167, 2009.

[111] F. Ozsolak, L. L. Poling, Z. Wang, H. Liu, X. S. Liu, R. G. Roeder, X. Zhang, J. S. Song, and D. E. Fisher. Chromatin structure analyses identify miRNA promoters. *Genes Dev.*, 22(22):3172–3183, 2008.

[112] D. Paez-Espino, G. A. Pavlopoulos, N. N. Ivanova, and N. C. Kyrpides. Nontargeted virus sequence discovery pipeline and virus clustering for metagenomic data. *nature protocols*, 12(8):1673, 2017.

[113] P. J. Park. Chip–seq: advantages and challenges of a maturing technology. *Nature Reviews Genetics*, 10(10):669–680, 2009.

[114] J. Peccoud, S. Lequime, I. Moltini-Conclois, I. Giraud, L. Lambrechts, and C. Gilbert. A Survey of Virus Recombination Uncovers Canonical Features of Artificial Chimeras Generated During Deep Sequencing Library Preparation. *G3: Genes, Genomes, Genetics*, 8(4):1129–1138, 2018.

[115] Y. Peng, H. C. Leung, S.-M. Yiu, and F. Y. Chin. Idba-ud: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, 28(11):1420–1428, 2012.

[116] Y. Peng, H. C. M. Leung, S. M. Yiu, and F. Y. L. Chin. Meta-IDBA: a de novo assembler for metagenomic data. *Bioinformatics*, 27(13):i94–i101, 2011.

[117] B. E. Pickett, E. L. Sadat, Y. Zhang, J. M. Noronha, R. B. Squires, V. Hunt, M. Liu, S. Kumar, S. Zaremba, Z. Gu, et al. ViPR: an open bioinformatics database and analysis resource for virology research. *Nucleic acids research*, 40(D1):D593–D598, 2011.

[118] S. Prabhakaran, M. Rey, O. Zagordi, N. Beerenwinkel, and V. Roth. Hiv-haplotype inference using a constraint-based dirichlet process mixture model. In *Machine Learning in Computational Biology (MLCB) NIPS Workshop*, pages 1–4, 2010.

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

[120] J. Qin, R. Li, J. Raes, M. Arumugam, K. S. Burgdorf, C. Manichanh, T. Nielsen, N. Pons, F. Levenez, T. Yamada, et al. A human gut microbial gene catalogue established by metagenomic sequencing. *nature*, 464(7285):59–65, 2010.

[121] S. Rajasekaran and M. Nicolae. An elegant algorithm for the construction of suffix arrays. *Journal of Discrete Algorithms*, 27:21–28, 2014.

[122] S. Rampelli, M. Soverini, S. Turroni, S. Quercia, E. Biagi, P. Brigidi, and M. Candela. ViromeScan: a new tool for metagenomic viral community profiling. *BMC genomics*, 17(1):165, 2016.

[123] G. Robertson, J. Schein, R. Chiu, R. Corbett, M. Field, S. D. Jackman, K. Mungall, S. Lee, H. M. Okada, J. Q. Qian, et al. De novo assembly and analysis of rna-seq data. *Nature methods*, 7(11):909–912, 2010.

[124] K. Rotmistrovsky and R. Agarwala. BMTagger: Best Match Tagger for removing human reads from metagenomics datasets. 2011.

[125] H. Rouha, C. Thurner, and C. W. Mandl. Functional microRNA generated from a cytoplasmic RNA virus. *Nucl. Acids Res.*, 38(22):8328–8337, 2010.

[126] S. Roux, J. R. Brum, B. E. Dutilh, S. Sunagawa, M. B. Duhaime, A. Loy, B. T. Poulos, N. Solonenko, E. Lara, J. Poulain, et al. Ecogenomics and potential biogeochemical impacts of globally abundant ocean viruses. *Nature*, 537(7622):689–693, 2016.

[127] S. Roux, F. Enault, B. L. Hurwitz, and M. B. Sullivan. VirSorter: mining viral signal from microbial genomic data. *PeerJ*, 3:e985, 2015.

[128] S. Roux, J. Tournayre, A. Mahul, D. Debroas, and F. Enault. Metavir 2: new tools for viral metagenome comparison and assembled virome analysis. *BMC bioinformatics*, 15(1):76, 2014.

[129] J. G. Ruby, P. Bellare, and J. L. DeRisi. Price: software for the targeted assembly of components of (meta) genomic sequence data. *G3: Genes| Genomes| Genetics*, 3(5):865–880, 2013.

[130] J. G. Ruby, C. H. Jan, and D. P. Bartel. Intronic microRNA precursors that bypass Drosha processing. *Nature*, 448(7149):83–86, 2007.

[131] H. K. Saini, A. J. Enright, and S. Griffiths-Jones. Annotation of mammalian primary microRNAs. *BMC genomics*, 9(1):564, 2008.

[132] H. K. Saini, S. Griffiths-Jones, and A. J. Enright. Genomic analysis of human microRNA transcripts. *Proc. Natl. Acad. Sci. USA*, 104(45):17719–17724, 2007.

[133] S. L. Salzberg, D. D. Sommer, D. Puiu, and V. T. Lee. Gene-boosted assembly of a novel bacterial genome from very short reads. *PLOS Comput Biol*, 4(9):e1000186, 09 2008.

[134] A. Sandelin, W. Alkema, P. Engström, W. W. Wasserman, and B. Lenhard. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucl. Acids Res.*, 32(suppl 1):D91–D94, 2004.

[135] A. Sandelin, P. Carninci, B. Lenhard, J. Ponjavic, Y. Hayashizaki, and D. A. Hume. Mammalian RNA polymerase II core promoters: insights from genome-wide studies. *Nature Rev. Genet.*, 8(6):424–436, 2007.

[136] M. Schirmer, W. T. Sloan, and C. Quince. Benchmarking of viral haplotype reconstruction programmes: an overview of the capacities and limitations of currently available programmes. *Briefings in bioinformatics*, page bbs081, 2012.

[137] D. R. Schoenberg and L. E. Maquat. Re-capping the message. *Trends Biochem. Sci.*, 34(9):435–442, 2009.

[138] D. E. Schones, K. Cui, S. Cuddapah, T.-Y. Roh, A. Barski, Z. Wang, G. Wei, and K. Zhao. Dynamic regulation of nucleosome positioning in the human genome. *Cell*, 132(5):887–898, 2008.

[139] J. S. Shapiro, R. A. Langlois, A. M. Pham, et al. Evidence for a cytoplasmic microprocessor of pri-miRNAs. *RNA*, 18(7):1338–1346, 2012.

[140] D. Sharma, P. Priyadarshini, and S. Vrati. Unraveling the web of viroinformatics: computational tools and databases in virus research. *Journal of virology*, 89(3):1489–1501, 2015.

[141] A. A. Sigova, A. C. Mullen, B. Molinie, S. Gupta, D. A. Orlando, M. G. Guenther, A. E. Almada, C. Lin, P. A. Sharp, C. C. Giallourakis, et al. Divergent transcription of long noncoding RNA/mRNA gene pairs in embryonic stem cells. *Proc. Natl. Acad. Sci. USA*, 110(8):2876–2881, 2013.

[142] J. T. Simpson and R. Durbin. Efficient de novo assembly of large genomes using compressed data structures. *Genome research*, 22(3):549–556, 2012.

[143] G. Song and L. Wang. MiR-433 and miR-127 arise from independent overlapping primary transcripts encoded by the miR-433-127 locus. *PLoS ONE*, 3(10):e3574, 2008.

[144] J. Spieth, D. Lawson, P. Davis, G. Williams, and K. Howe. Overview of gene structure in C. elegans. *WormBook*, .:1–18, 2014.

[145] S. R. A. S. Staff. Using the sra toolkit to convert .sra files into other formats. In: SRA Knowledge Base [Internet], 2011. Available at: `http://www.ncbi.nlm.nih.gov/books/NBK158900/`. (Accessed: 25 August 2015).

[146] S. T O'Neil and S. J. Emrich. Haplotype and minimum-chimerism consensus determination using short sequence data. *BMC genomics*, 13(2):S4, 2012.

[147] A. Töpfer, T. Marschall, R. A. Bull, F. Luciani, A. Schönhuth, and N. Beerenwinkel. Viral quasispecies assembly via maximal clique enumeration. *PLoS Comput Biol*, 10(3):e1003515, 2014.

[148] A. Töpfer, O. Zagordi, S. Prabhakaran, V. Roth, E. Halperin, and N. Beerenwinkel. Probabilistic inference of viral quasispecies subject to recombination. *Journal of Computational Biology*, 20(2):113–123, 2013.

[149] C. Trapnell and S. L. Salzberg. How to map billions of short reads onto genomes. *Nature biotechnology*, 27(5):455, 2009.

[150] T. Treangen, S. Koren, D. Sommer, B. Liu, I. Astrovskaya, B. Ondov, A. Darling, A. Phillippy, and M. Pop. MetAMOS: a modular and open source metagenomic assembly and analysis pipeline. *Genome Biology*, 14(1):R2, 2013.

[151] D. T. Truong, E. A. Franzosa, T. L. Tickle, M. Scholz, G. Weingart, E. Pasolli, A. Tett, C. Huttenhower, and N. Segata. Metaphlan2 for enhanced metagenomic taxonomic profiling. *Nature methods*, 12(10):902, 2015.

[152] D. T. Truong, A. Tett, E. Pasolli, C. Huttenhower, and N. Segata. Microbial strain-level population structure and genetic diversity from metagenomes. *Genome research*, 2017.

[153] M. Vignuzzi, J. K. Stone, J. J. Arnold, C. E. Cameron, and R. Andino. Quasispecies diversity determines pathogenesis through cooperative interactions in a viral population. *Nature*, 439(7074):344–348, 2006.

[154] G. Wang, Y. Wang, C. Shen, Y.-w. Huang, K. Huang, T. H. Huang, K. P. Nephew, L. Li, and Y. Liu. RNA polymerase II binding patterns reveal genomic regions involved in microRNA gene regulation. *PLoS ONE*, 5(11):e13798, 2010.

[155] Z. Wang, M. Gerstein, and M. Snyder. Rna-seq: a revolutionary tool for transcriptomics. *Nature reviews genetics*, 10(1):57–63, 2009.

[156] P. Willeit, P. Skroblin, S. Kiechl, C. Fernández-Hernando, and M. Mayr. Liver micrornas: potential mediators and biomarkers for metabolic and cardiovascular disease? *European heart journal*, page ehw146, 2016.

[157] J. Winter, S. Jung, S. Keller, R. I. Gregory, and S. Diederichs. Many roads to maturity: microRNA biogenesis pathways and their regulation. *Nature Cell Biol.*, 11(3):228–234, 2009.

[158] M. E. Woolhouse, A. Rambaut, and P. Kellam. Lessons from Ebola: Improving infectious disease surveillance to inform outbreak management. *Science translational medicine*, 7(307):307rv5–307rv5, 2015.

[159] Y. Wu, M. Rho, T. G. Doak, and Y. Ye. Stitching gene fragments with a network matching algorithm improves gene assembly for metagenomics. *Bioinformatics*, 28(18):i363–i369, 2012.

[160] Y.-W. Wu, Y.-H. Tang, S. G. Tringe, B. A. Simmons, and S. W. Singer. MaxBin: an automated binning method to recover individual genomes from metagenomes using an expectation-maximization algorithm. *Microbiome*, 2(1):26, 2014.

[161] M. Xie, M. Li, A. Vilborg, N. Lee, M.-D. Shu, V. Yartseva, N. Šestan, and J. a. Steitz. Mammalian 5'-capped microRNA precursors that generate a single microRNA. *Cell*, 155(7):1568–1580, Dec. 2013.

[162] A. Yamashita, T. Sekizuka, and M. Kuroda. VirusTAP: viral genome-targeted assembly pipeline. *Frontiers in microbiology*, 7:32, 2016.

[163] C. Yuan, J. Lei, J. Cole, and Y. Sun. Reconstructing 16s rrna genes in metagenomic data. *Bioinformatics*, 31(12):i35–i43, 2015.

[164] C. Yuan and Y. Sun. RNA-CODE: A Noncoding RNA Classification Tool for Short Reads in NGS Data Lacking Reference Genomes. *PLoS ONE*, 8:e77596, 10 2013.

[165] N. Yutin, K. S. Makarova, A. B. Gussow, M. Krupovic, A. Segall, R. A. Edwards, and E. V. Koonin. Discovery of an expansive bacteriophage family that includes the most abundant viruses from the human gut. *Nature microbiology*, 3(1):38, 2018.

[166] O. Zagordi, A. Bhattacharya, N. Eriksson, and N. Beerenwinkel. Shorah: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC bioinformatics*, 12(1):119, 2011.

[167] J. Zhang, K. Kobert, T. Flouri, and A. Stamatakis. Pear: a fast and accurate illumina paired-

end read merger. *Bioinformatics*, 30(5):614–620, 2014.

[168] Y. Zhang, T. Liu, C. A. Meyer, J. Eeckhoute, D. S. Johnson, B. E. Bernstein, C. Nusbaum, R. M. Myers, M. Brown, W. Li, et al. Model-based analysis of ChIP-Seq (MACS). *Genome Biol.*, 9(9):R137, 2008.

[169] Y. Zhang, Y. Sun, and J. R. Cole. A scalable and accurate targeted gene assembly tool (sat-assembler) for next-generation sequencing data. *PLoS Comput Biol*, 10(8):e1003737, 2014.