Bioinformatics, 34(17), 2018, 2927–2935
doi: 10.1093/bioinformatics/bty202
Advance Access Publication Date: 3 April 2018
Original Paper

OXFORD

Sequence analysis

De novo haplotype reconstruction in viral quasispecies using paired-end read guided path finding

Jiao Chen[1], Yingchao Zhao[2] and Yanni Sun[1,*]

[1]Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA and [2]School of Computing and Information Sciences, Caritas Institute of Higher Education, Hong Kong, 999077, China

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on August 28, 2017; revised on March 12, 2018; editorial decision on March 25, 2018; accepted on April 2, 2018

Abstract

Motivation: RNA virus populations contain different but genetically related strains, all infecting an individual host. Reconstruction of the viral haplotypes is a fundamental step to characterize the virus population, predict their viral phenotypes and finally provide important information for clinical treatment and prevention. Advances of the next-generation sequencing technologies open up new opportunities to assemble full-length haplotypes. However, error-prone short reads, high similarities between related strains, an unknown number of haplotypes pose computational challenges for reference-free haplotype reconstruction. There is still much room to improve the performance of existing haplotype assembly tools.

Results: In this work, we developed a *de novo* haplotype reconstruction tool named PEHaplo, 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.

Availability and implementation: The source code and the documentation of PEHaplo are available at https://github.com/chjiao/PEHaplo.

Contact: yannisun@msu.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

High mutation rate, natural selections and recombinations can lead to high genetic diversity of RNA virus populations (Domingo-Calap *et al.*, 2016), which consist of closely related but different viral strains. These groups of virus populations are often termed as viral quasispecies (Nowak, 2006). Each strain in the quasispecies is defined by its haplotype sequence. Commonly known examples of the fast mutating viruses include clinically important viruses such as human immunodeficiency virus (HIV-1) and the hepatitis C virus (HCV). The genetic heterogeneity of the virus populations is the key to their adaptive behavior. As the natural selection works on a set of sequences rather than one, high genetic diversity confers the viruses the abilities to escape host immune responses or develop drug resistance. Reconstruction of the viral haplotypes is a fundamental step to characterize the structure of the virus populations, predict viral phenotypes and finally provide important information for clinical treatment and prevention (Schirmer *et al.*, 2014).

Development of next-generation sequencing technologies advances the characterization of the haplotypes and their abundance in heterogeneous virus populations. The deep sequencing of virus population samples becomes available and various methods and tools have been devised for viral haplotype reconstruction (Baaijens *et al.*, 2017; Jayasundara *et al.*, 2015; Malhotra *et al.*, 2015; Mangul *et al.*, 2014; Töpfer *et al.*, 2014). The methods can be

divided into two groups based on their dependency on a reference genome (Beerenwinkel *et al.*, 2012). The first group of methods needs reference genomes and employs read alignments against the reference sequence to infer haplotypes. However, due to the high mutation rate, high quality reference genomes of virus populations are not always available. In particular, for emerging infectious viral diseases such as SARS, which lack reference genomes during the breakout, reference-based methods are not plausible. The second group of methods belongs to *de novo* haplotype reconstruction, which does not require reference genomes. These methods can characterize new viral strains or novel haplotypes. Our work belongs to the second group.

A recent review of chosen haplotype reconstruction tools has shown that haplotype recovery is a computationally challenging problem (Schirmer *et al.*, 2014). The authors' benchmarking results on a series of data demonstrated the performance of the tested programs is poor when sequence divergence is low. In addition, these programs failed to recover rare haplotypes. Thus, there is a pressing need for new methods and tools for more accurate haplotype reconstruction.

The tested programs in the review (Schirmer *et al.*, 2014) all belong to group 1, which require alignments against a reference sequence as input. Without using a reference sequence, our method of haplotype reconstruction from deep sequencing data uses a method similar to *de novo* genome assembly. Applying assembly to viral haplotype reconstruction faces several challenges. The first challenge is to distinguish highly similar genomes of different strains. Supplementary Figure S3 and Table S1 show the high sequence similarity between the genomes of interest. Existing assembly methods tend to produce either short or chimeric contigs for deep sequencing data containing highly similar genomes. The second challenge is the extreme difficulty in distinguishing sequencing errors from mutations of a rare haplotype. The third challenge is to recover the haplotypes of low abundance.

1.1 Related work

There exist a number of metagenomic assembly tools that could be applied to assemble viral genomes in quasispecies data (Laserson *et al.*, 2011; Namiki *et al.*, 2012; Peng *et al.*, 2011; Salzberg *et al.*, 2008; Treangen *et al.*, 2013). However, these assembly tools cannot distinguish different haplotypes and only produce fragmented or chimeric contigs.

There are a group of tools specifically designed for viral haplotype reconstruction (Astrovskaya *et al.*, 2011; Baaijens *et al.*, 2017; Huang *et al.*, 2011; Jayasundara *et al.*, 2015; Malhotra *et al.*, 2013; Malhotra *et al.*, 2015; Mangul *et al.*, 2014; Prabhakaran *et al.*, 2010; Prabhakaran *et al.*, 2014; Prosperi and Salemi, 2012; T O'neil and Emrich, 2012; Töpfer *et al.*, 2013; Töpfer *et al.*, 2014; Zagordi *et al.*, 2011). Of them, HaploClique, MLEHaplo and SAVAGE were recently published and are closely related to our method. They all utilized the paired-end reads. HaploClique uses the insert size distribution for detection of large indels and clique enumeration to distinguish mutations from sequencing errors. However, it needs a reference sequence for generating alignment graphs. HaploClique provides a source of inspiration for SAVAGE, which is the first tool for *de novo* assembly of viral haplotypes using overlap graphs. SAVAGE joins overlapped read pairs before merging short reads using cliques. It has been benchmarked with other virus assembly tools and showed its superiority over other tools in a comprehensive set of assembly metrics. MLEHaplo also explicitly employs paired-end reads for finding top-score paths. MLEHaplo and our method are based on two different graph models: de Bruijn graph and overlap graph. The major differences of these two types of graphs are detailed in a review paper (Li *et al.*, 2012). We chose error correction tool that is specifically designed for overlap graphs. Both tools applied topology-based graph pruning. The graph pruning techniques that are unique to PEHaplo will be described in Section 2.2.2. In addition, during path finding, we carefully distinguish paired-end connections formed by different types of nodes in order to improve the accuracy of path finding. Focusing on *de novo* assembly tools, we will benchmark our work against SAVAGE and MLEHaplo.

In this work, we designed and implemented PEHaplo, which assembles viral haplotypes from deep sequencing data. We created a novel overlap graph incorporating paired-end reads information, which is utilized in graph pruning, path finding and contig refinement. PEHaplo was applied to both simulated and real viral deep sequencing data and was benchmarked with the recently published tools. The experimental results show that PEHaplo can recover viral haplotypes with longer contigs and higher accuracy.

2 Materials and methods

A major challenge for viral haplotype reconstruction is the high sequence similarity between viral strains. In particular, the distribution of the mutations/insertions/deletions largely determines the difficulty levels of the problem. Here, longest common substring (LCS) is used to refer to the LCS between any two neighboring mutations/insertions/deletions. Depending on the size of the LCS, we have three cases as shown in Figure 1.

- If LCS size $\leq$ read size, haplotype reconstruction can be solved based on read overlaps (Fig. 1A).
- If LCS size $\geq$ read size but $\leq$ insert size, paired-end reads are able to distinguish different haplotypes (Fig. 1B).
- If LCS size $\geq$ insert size, coverage information can be utilized to distinguish haplotypes of different abundance (Fig. 1C).

In order to classify viral haplotype reconstruction problems in the above three cases, it is necessary to know the LCS distributions inside each quasispecies. While the insert size can be estimated for different sequencing platforms, it is not feasible to empirically obtain all viral strains and compute the sizes of their LCSs. We thus rely on quasispecies theory (Nowak, 2006) for estimating the LCS sizes using an average viral mutation rate. The detailed method and also the generated distribution of LCSs can be found in Supplementary Material Section S1.

According to our computed LCS distribution, the LCS sizes span all three cases in Figure 1. Therefore, our methods use three types of information for virus assembly. (i) Paired-end reads. As paired-end reads are sequenced from the same fragment, they thus belong to the same haplotype. (ii) Coverage. If two strains have highly different

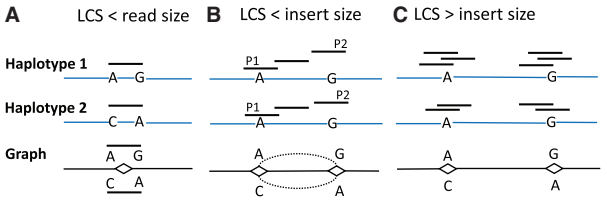

Fig. 1. Distinguishing two haplotypes with different LCS sizes. In panel **B**, P1 and P2 represent the ends of a read pair. The problem becomes harder with increase in the LCS sizes

coverages, they can be distinguished using coverage information. (iii) Enumeration of cliques. Reads forming cliques in the overlap graph tend to come from the same haplotype. Several recently published tools (Baaijens *et al.*, 2017; Töpfer *et al.*, 2014) employed clique enumeration for haplotype reconstruction. Although paired-end reads have been previously employed in haplotype reconstruction, they were not carefully elaborated and analyzed. We conducted a deep analysis of the utility and limitations of using paired-end reads for haplotype reconstruction.

We organize the Section 2 as follows: first introducing the paired-end overlap graph and the key idea of path finding using paired-end reads, then outlining the complete pipeline and describing the major components.

2.1 Paired-end overlap graph and path finding

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.

In our method, a paired-end overlap graph (PE_G), which adds information from paired end reads to a standard overlap graph, has been constructed. PE_G has the same node set as an overlap graph but has two sets of edges. One set of edges are inherited from a standard overlap graph. The other set of edges connect nodes whose reads form paired-end reads. Intuitively, while an overlap graph records the connectivity between reads, PE_G also records the number of paired-end reads between nodes. Thus, PE_G can be defined as $G(V, E, E')$, where E is the same as in an overlap graph. If two reads form a paired-end read pair, an edge in set E' is created between the corresponding nodes.

Figure 2 shows an example of paired-end overlap graph. Figure 2A contains reads sequenced from two strains that differ by only two mutations. The overlap threshold is set as half of the read size. The corresponding paired-end overlap graph PE_G is shown in (B). The edges in E are plotted by solid lines and the edges in E' by dashed lines. Nodes a.1 and a.2 are a read pair and thus form an edge in E'. Similarly, nodes d.1 and d.2 have an edge in E' because d.1 and d.2 are a read pair.

In the graph, there are four complete paths: $a.1 \to b \to c \to e \to f \to a.2$, $a.1 \to b \to c \to e \to f \to d.2$, $d.1 \to b \to c \to e \to f \to a.2$ and $d.1 \to b \to c \to e \to f \to a.2$. The goal of the assembly step is to output the two correct paths (i.e. $a.1$ to $a.2$ and $d.1$ to $d.2$), using the guidance from edges in E'. Specifically, for a path starting with a.1, the dashed edges in E' ($a.1 \to a.2$) will guide the path to the correct node a.2. Similarly, a path starting with d.1 will end with d.2 based on the guidance of the dashed edge $d.1 \to d.2$. Thus, the path finding using the paired-end information will output two paths, representing the two haplotypes.

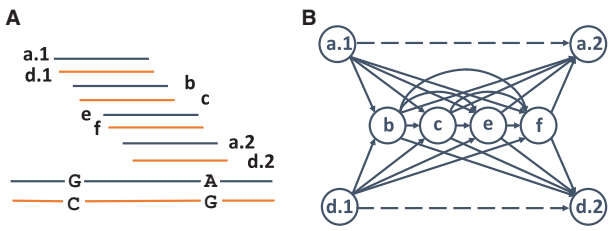

Fig. 2. (**A**) The bottom two long lines represent two haplotypes, which only differ by two mutations at two loci (G-C and A-G). Short lines represent reads sequenced from the two strains. The reads are sorted by their read mapping positions against their native strain. a.1 and a.2 are a read pair from the thick strain. d.1 and d.2 are the read pair from the thin strain. (**B**) Paired-end overlap graph. Nodes b, c, e, and f originate from the common region of the two strains. The dashed lines represent paired-end read connection

2.2 The whole pipeline of PEHaplo

There are five major components in the pipeline of PEHaplo as shown in Figure 3. In the first pre-processing stage, reads with 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 tool Karect (Allam *et al.*, 2015). We chose Karect because it was more recently published and also it takes advantage of the whole reads rather than k-mers for error correction. Duplicated reads and substring reads are then removed from the corrected reads. The detailed pre-processing description can be found in the Supplementary Material Section S2. Second, an overlap graph is built from the pre-processed reads with overlaps computed by Readjoiner (Gonnella and Kurtz, 2012) and the strand of reads is adjusted by traversing the graph. The detailed strategy about strand adjustment can be found in Supplementary Material Section S3. The output reads will have the same orientation. The third stage will build the overlap graph again from the strand-adjusted reads with overlaps computed by Apsp (Haj Rachid and Malluhi, 2015) and utilize various graph pruning methods to remove possible random overlaps and simplify the graph for efficient assembly. In the fourth stage, E' will be constructed and paired-end guided path finding algorithms are applied to produce contigs from the paired-end overlap graph. Finally, the paired-end reads are aligned against produced contigs by bowtie2 (Langmead and Salzberg, 2012) to identify and correct potential mis-join errors.

2.2.1 Paired-end overlap graph construction

There are two steps in construction of the paired-end overlap graph. In the first step, the standard overlap graph G is constructed, followed by collapsing nodes and merging cliques. In the second step, the numbers of paired-end reads between nodes in G are identified and false connected edges are removed using added paired-end information. This section will focus on the overlap computation between reads.

All remaining reads 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 datasets. There are efficient implementation of all-pairs suffix–prefix comparison algorithms based on data structures such as hashing table or compact prefix tree (Gonnella and Kurtz, 2012; Haj Rachid and Malluhi, 2015). In PEHaplo, we first compute all

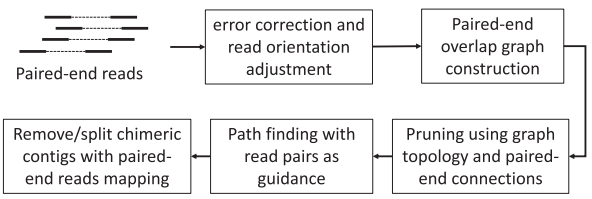

Fig. 3. The pipeline and main components of PEHaplo. Note that the error correction component is implemented using Karect (Allam *et al.*, 2015), which uses alignments between reads rather than alignments between reads and a reference genome for error correction. The read orientation adjustment uses the overlap results from Readjoiner (Gonnella and Kurtz, 2012), and the paired-end overlap graph is built based on overlaps computed by Apsp (Haj Rachid and Malluhi, 2015)

suffix–prefix matches between reads for read orientation adjustment and then construct the final overlap graph from adjusted reads.

Overlap cutoff estimation: The overlap cutoff l is an important parameter. A small l tends to keep most true overlaps but also introduces false edges. A large l is likely to eliminate most false overlaps but can possibly miss true connections for reads from lowly sequenced regions. In our methods, an exponential distribution is used to estimate the appropriate overlap cutoff.

Let N be the total reads number, r be the read length, and L be the genome size, the sequencing coverage is calculated as $C = Nr/L$. We use the Poisson distribution to model the number of reads sequenced from unit length of a genome. The parameter λ in the Poisson distribution is estimated as N/L (Jiang and Wong, 2009). The distance (X) (see Fig. 4) between two adjacent reads will thus follow an exponential distribution. The corresponding cumulative distribution function is the probability that two adjacent reads have an overlap size of at least $r - d$, where d is a given upper bound of X. For example, Let $r = 250$, $L = 10\ 000$ and $N = 800$. Then the sequencing coverage C is 20 and $\lambda(N/L)$ is 0.08. Thus, for given d as 70, we have $F(X \leq 70) = 0.9963$. That is, there is 99.63% possibility that two adjacent reads will form an overlap with size of at least 180 (i.e. 250–70). The expected number of adjacent read pairs with overlap less than $r - d$ can be calculated as $(N-1)e^{-\lambda d}$. Since there are 800 reads, the expected number of pairs with overlap smaller than 180 is: $799 * (1 - 0.9963) = 2.96$. Therefore, four contigs may be produced for the genome under this overlap cutoff. Our empirical experience shows that choosing an overlap threshold with $(N-1)e^{-\lambda d}$ being around 0.01 usually produces good assembly results. While keeping the connectivity of the graph, the overlap threshold should be as large as possible.

2.2.2 Graph pruning

The original overlap graph is often very complex because of the large data size, transitive edges, sequencing errors and highly similar regions shared by haplotypes. After transitive reduction, the node collapsing operation is applied. For an edge $u \rightarrow v$, if the out-degree of u and the in-degree of v are both 1, they can be merged into a new node without losing the connectivity of the graph. This collapsing operation is applied iteratively on the whole graph until there is no such edge (Yuan *et al.*, 2015). After transitive reduction and node collapsing, an iterative graph pruning procedure is applied to repeatedly simplify the graph at each iteration.

Merging cliques: We are interested in cliques in the overlap graph because reads within a clique can share true mutations while sequencing errors are usually random and are not shared by other reads. Therefore, cliques can be used to distinguish true mutations from sequencing errors. In PEHaplo, we prune the graph by removing the edges from nodes inside of cliques to nodes outside of cliques. The details can be found in Supplementary Material Section S4.

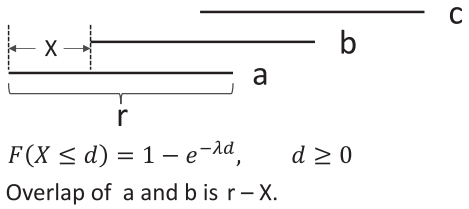

Fig. 4. The distance between two adjacent reads can be estimated by an exponential distribution. *a*, *b*, *c* represent three reads. *r* is the read length

Removing false edges using read pairs: Due to the nature of viral 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 the same origin of the two reads. Wrong edges increase the complexity of the graph and may also produce misjoined contigs. Paired-end information is employed to remove potentially wrong edges.

There are two key observations about the edges formed by reads from different strains. First, the end nodes usually have other incident edges incurred by the correction connections. Second, the wrong edge or the contigs containing the wrong edge are not well supported by read pairs. Therefore, paired-end information is used as evidence to remove false edges. Each edge formed by nodes with large in or out degrees (Cormen, 2009) is examined because of the significant chance that some of the incident edges are false overlaps.

Figure 5A presents an example. Edge $u \rightarrow v$ is one of the many edges incident to nodes u or v. The following rules for edge $u \rightarrow v$ are applied: if there is no read pair support between u and v, between u's predecessor nodes and v, or between u and v's successor nodes, and the sequence formed by joining u, v is longer than an insert size cutoff, remove $u \rightarrow v$. The insert size cutoff can be customized depending on the given data properties. To remove false connected edges, each node and edge of the overlap graph need to be traversed. The time complexity is $O(V + E)$.

2.2.3 Paired-end guided path finding

Once the overlap graph is pruned, paired-end connections will be added and thus form the paired-end overlap graph PE_G. As a quick review, PE_G is defined as (V, E, E'), where E contains the edges from read overlaps while E' contains edges from paired-end connections. The weight of an edge in E' represents the number of paired-end read pairs between two end nodes. Note that after node collapse, each node can contain multiple reads.

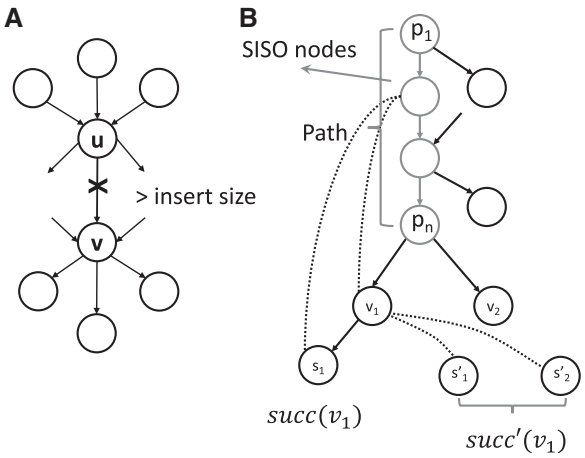

Fig. 5. (**A**) Removing false edges using paired-end information. The overlap edges with 'X' will be removed if insufficient read pairs exist between their ends. (**B**) In this example, the ending node p_n of current path has two successors v_1 and v_2. The path and a SISO nodes in the path are marked. The solid lines are overlaps and dashed lines are paired-end connections between nodes. Six scores are computed to select the right successor for extending the path. Among these, five are calculated as paired-end edge weights between nodes in current path and nodes associated with the successor. As for v_1, succ(v_1) and succ$'$(v_1) are shown in the figure. The group 1 scores contain the paired-end edge weights between the SISO node and v_1, s_1, and s'_1, s'_2. Group 2 scores contain the paired-end edge weights between path nodes $\{p_1, \ldots, p_{n-1}\}$ and v_1, s_1

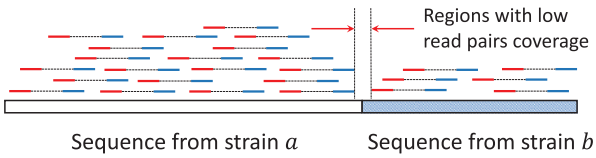

Fig. 6. Read pairs mapping profile on a misjoined contig. The contig is shown as the long bar at the bottom, which is misjoined with two sequences from strains *a* and *b*. The dashed line connects the two ends of a read pair. Fewer read pairs will go across the misjoined location, thus revealing a valley in the aligned reads profile, which can be used to split the contig

The problem of assembling a single haplotype in the graph PE_G can be formulated as finding a path p, so that p's weight defined using edges in E' is maximized. Intuitively, we look for paths with the best support of paired-end connections. This process can be repeated to find k longest paths. Unfortunately, finding the path with the greatest number of paired-end connections in PE_G is NP-complete . The detailed proof can be found in the Supplementary Material Section S5.

Thus, we designed a greedy path finding algorithm for path extension. At each vertex with multiple successors, the greedy algorithm chooses the best node for extension. It carefully considers paired-end connections between different types of nodes and also the coverage information. Paired-end connections between nodes can be efficiently accessed from the constructed paired-end graph PE_G.

Scores calculation for path extension: To find correct paths from the graph, the right node needs to be selected each time we extend the path. In particular, when a node has multiple successors, a right choice must be made for path extension. In general, a successor with the most paired-end connections is chosen to the current path. Different types of paired-end connections were treated with different priorities in distinguishing haplotypes; in particular, single-in single-out (SISO) nodes are differentiated from other nodes. After error correction and node collapsing, SISO nodes, which have in-degree of 1 and out-degree of 1, tend to represent one haplotype (Supplementary Fig. S4). Any paired-end connection incident to SISO nodes can be used to recruit nodes that belong to the same haplotype. Other nodes originating from common regions of two or more haplotypes usually have multiple child or parent nodes. Paired-end connections to those nodes do not provide useful guidance in path finding. In our algorithm, we distinguish paired-end connections involving SISO nodes and other nodes.

For $e' \in E'$, $e'(u, v)$ is the edge weight between u and v in E'. Let Path $= \{p_1, p_2, \ldots, p_n\}$ be the current path. The ending node p_n in the current path has multiple successors. As we have two sets of edges in PE_G, let succ(v) in E represents v's successor nodes in the standard overlap graph. succ$'$(v) in E' represents all nodes that form paired-end connections with v. For each successor v of p_n ($v \in$ succ(p_n)), we compute six scores, which are divided into three groups. An example of the scores calculation is shown in Figure 5B. Group 1 contains the paired-end edge weights between SISO nodes in Path and v (score 1), v's successor nodes in E (score 2), and v's successor nodes in E' (i.e. all nodes forming paired-end connections with v, score 3). Group 2 contains the paired-end edge weights between all path nodes (except p_n) and v (score 4), v's successor nodes in E (score 5). The third group contains coverage difference between *Path* and v (score 6).

The pseudocode in Supplementary Material Section S7 describes the greedy algorithm, which chooses the locally optimal node for path extension based on the above scores.

2.2.4 Correcting contigs with paired-end read distribution

To further improve the quality of assembled contigs, we apply a contig correction method similar to the tool PECC (Li *et al.*, 2017). With the contigs generated after path finding, the raw reads are aligned to them and contigs are split from the locations with low read pairs coverage (Fig. 6).

3 Results

In this section, we will evaluate the performance of PEHaplo on both simulated and real viral quasispecies datasets. The simulated dataset includes both the commonly adopted HIV dataset and also highly biased dataset with rare haplotypes. For each experiment, we will present the performance of PEHaplo and benchmark it with recently published *de novo* quasispecies assembly tools. In addition, we carefully evaluate the performance of each main component in the whole pipeline. The results show that our tool produces fewer and significantly longer contigs which recover a majority of the haplotypes.

3.1 Experimental datasets

We evaluated PEHaplo on several simulated datasets, one real HIV-1 Illumina MiSeq sequencing dataset, and one real Influenza sample. Both the simulated and real HIV-1 datasets 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% (Supplementary Table S1). HXB2 and NL43 have the highest similarity with LCS of size 427 bp (Supplementary Table S2). We choose HIV because it is adopted by other viral haplotype reconstruction tools and has become a leading standard for performance evaluation.

3.2 Evaluation metrics

The haplotype sequences and compositions are known in these datasets and we are able to evaluate the quality of the assembled contigs generated by all tools. The produced results were compared to recently published *de novo* assembly tools IVA (Hunt *et al.*, 2015), MLEHaplo (Malhotra *et al.*, 2015) and SAVAGE (Baaijens *et al.*, 2017).

Following SAVAGE, a third-party tool MetaQuast (Mikheenko *et al.*, 2015) is used for evaluating the output of all tested tools. MetaQuast integrated several components for convenient *de novo* assembly performance evaluation. It aligns the generated contigs to the viral reference genomes and reports the number of contigs, N50, unaligned length, target genome(s) covered, mismatch and indel rates and so on. Only the best matching position for each contig is reported. N50 length is defined as the maximum length in which all contigs of at least this length contain at least 50% of all the contig bases. A contig can be partially aligned to a reference sequence. Thus, the total length of all unaligned parts is reported as 'unaligned length'. For the aligned parts, 'target genome(s) covered' and 'mismatch and indel rates' are computed. Target genome(s) covered is the percentage of reference genomes that are aligned by contigs, and mismatch/indel rate is the percentage of mismatches/indels of aligned contigs.

3.3 Results on HIV simulated dataset

PEHaplo was first applied on a simulated HIV-1 quasispecies dataset. We used ART-illumina (Huang *et al.*, 2012) to simulate 1.9e + 5 paired-end, 250 bp error-containing MiSeq reads from the five HIV-1 strains with average insert size of 600 bp and standard deviation

of 150 bp. The total coverage of the five strains is ∼5000x, which is close to the coverage of real HIV quasispecies data commonly used by existing tools. To obtain a more realistic dataset, a fitness-based power law equation (Barbosa *et al.*, 2012) was used to simulate the coverage distribution among five strains: $C_i = bf_i^a, \quad i = 1, 2, \ldots, 5$, where C_i and f_i denote the coverage and fitness of strain i, respectively. The coverages for each strain in the simulated dataset are as follows: 89.6–2190×, HXB2–1095×, JRCSF–730×, NL43–547× and YU2–438×.

Following the PEHaplo pipeline, we first performed error correction and duplicated sequence removal on the raw simulated dataset. With 1.9e + 5 error corrected reads, 48 833 reads were kept after removing duplicates. Only those reads that duplicate at least three times in the raw data were kept, further reducing the reads number to 26 961. After adjusting reads orientation, an overlap graph was constructed with the tool Apsp (Haj Rachid and Malluhi, 2015). The original overlap graph has 26 961 nodes and 977 570 edges. After merging cliques, removing transitive edges and collapsing nodes, 63 nodes and 67 edges were left. We then applied false edge removal and node collapsing on the graph, and further reduced it to 48 nodes and 44 edges. Paths and contigs were generated from this pruned graph using the greedy algorithm described in the Section 2.

To evaluate the effectiveness of our false edge removal step, the simulated reads was also assembled without the false edges removal step. The results are shown in Supplementary Table S3. Comparing to contigs with false edge removal, the genome covered fraction is reduced from 97% to 91.8%. Also, the N50 value is decreased while the mismatch rate is increased. The results reveal that our false edge removal step is effective and improves the final results.

PEHaplo generated 10 contigs from the simulated dataset and the results are summarized in Table 1. The contigs are able to cover over 97% on the five viral strains, with a N50 of 9274 bp. The largest contig has a length of 9668 bp, which almost covers a complete HIV strain. Meanwhile, these contigs have low mismatch and indel rates.

We also assembled the simulated reads with IVA, MLEHaplo and SAVAGE and summarized their results in Table 1. 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 the strain 89.6 and about 43% of the strain HXB2. The results of IVA real that it tends to generate one consensus genome sequence corresponding to the haplotype with the highest coverage. Other strains are largely missed. 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 the longest contig of 1716 bp. Following the guidance of SAVAGE tutorial, we set the overlap cutoff as 180 bp. SAVAGE produced 64 contigs covering 97% of the reference genomes, with a N50 of 1926 bp and the largest contigs of 7941 bp.

Comparing to IVA and MLEHaplo, PEHaplo is able to produce longer contigs with fewer mismatches and indels on the simulated HIV-1 dataset. SAVAGE also shows better performance than IVA and MLEHaplo on this dataset, but it produced many short contigs (N50 value 1926 bp versus 9262 bp of PEHaplo). Some of them cannot be aligned to the reference genomes, leading to larger 'unaligned length'.

Table 1. Assembly results on simulated HIV dataset for IVA, MLEHaplo, SAVAGE and PEHaplo. Contigs that are at least 500 bp are aligned to the reference haplotype sequences with a similarity cutoff of 98%

Tools	Contigs num	N50	Genomes covered (%)	Unaligned length (bp)	Mismatch rate (%)	Indels (%)
IVA	1	13 434	28.7	**0**	0.809	0.051
MLEHaplo	205	671	78.0	81 125	0.542	0.008
SAVAGE	64	1926	**97.32**	4792	**0.009**	0.004
PEHaplo	10	9274	97.0	**0**	0.026	**0.002**

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

The greedy algorithm carefully utilizing paired-end information plays a crucial role for producing high quality long contigs. In this section, we focus on evaluating the performance of path finding and investigating whether the improved performance of PEHaplo is simply due to the pruned graph or the combination of the pruned graph and path finding algorithm. Thus, we applied popular *de novo* metagenomic assembly tools IDBA-UD (Peng *et al.*, 2012) and Ray Meta (Boisvert *et al.*, 2012) on the pruned overlap graph that was used as the input for path finding in PEHaplo. In addition, as SAVAGE is the newest viral haplotype reconstruction tool and has better performance than IVA and MLEHaplo, we also applied SAVAGE on the same pruned overlap graph as PEHaplo.

With the same input reads or graph, we compared the output of these tools using MetaQuast and presented the results in Supplementary Table S4. The results revealed those contigs assembled by IDBA-UD, Ray Meta and SAVAGE from the reduced overlap graph are fragmented and cover only insufficient proportion of the five reference genomes. These contigs have low rate of mismatches and indels, but their average lengths are much shorter than PEHaplo. The experiments show that the paired-end guided path finding algorithm in PEHaplo is essential for producing long haplotype segments from the viral quasispecies sequencing data.

3.4 Benchmark on HIV MiSeq dataset

To further assess the performance of assembly methods, we applied PEHaplo on a real HIV quasispecies dataset (SRR961514), sequenced from the mix of five HIV-1 strains with Illumin MiSeq sequencing technology (Di Giallonardo *et al.*, 2014). This dataset contains 714 994 pairs (2 × 250 bp) of reads that cover the five strains to 20 000×.

Similar pre-processing procedures were performed 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 dataset has extremely high coverage on the five strains, we still kept those reads that duplicate at least three times in the raw dataset. After these pre-processing procedures, 26 691 reads were kept for strand adjustment and assembly.

PEHaplo produced 24 contigs from the real MiSeq HIV dataset that can cover over 92% of the five HIV-1 strains. These contigs have a N50 value of 2223 bp and the longest contig is 9133 bp. The results are summarized in Table 2. Compared to simulated HIV dataset, PEHaplo has generated more contigs but with a lower N50 value and higher mismatches and indels on the real dataset. We notice that the real HIV dataset contains more sequencing errors and has a larger variation for insert size than the simulated dataset.

The performance of PEHaplo was again compared with IVA, MLEHaplo and SAVAGE. IVA generated 10 contigs that can cover about 20% of the five strains. Similar to the simulated dataset, these contigs still cover larger parts on haplotypes with higher sequencing

Table 2. Assembly results on real HIV MiSeq dataset for IVA, MLEHaplo, SAVAGE and PEHaplo

Tools	Contigs num	N50	Genomes covered (%)	Unaligned length (bp)	Mismatch rate (%)	Indels (%)
IVA	10	1150	20.1	1150	0.660	0.052
MLEHaplo	234	6501	53.6	786 272	0.588	**0.035**
SAVAGE	846	588	92.6	**0**	0.161	0.040
PEHaplo	24	2223	**92.98**	**0**	**0.016**	0.045

coverage. With the same parameters as before, MLEHaplo produced 234 contigs that can cover over 53% of the five genomes with mismatch and indel rates similar to the simulated dataset. It generated much longer contigs on the real data, with a N50 value of 6501 bp and the largest contig of 8470 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 (Baaijens *et al.*, 2017) has shown results on the same dataset, the metrics in their literature are used for evaluation. From their results, SAVAGE produced 846 contigs covering over 92% of the reference genomes, with a N50 of 588 bp, and largest contig of 1221 bp (Table 2).

On the real HIV dataset, PEHaplo can still produce longer contigs with fewer mismatches than all three benchmarked tools. Overall, PEHaplo is able to assemble short reads sequenced from multiple viral strains sharing high similarities, generating long, high quality contigs that can reconstruct most of the target haplotypes. In Supplementary Figure S5, we show the contig alignment result on HXB2 strain for PEHaplo and SAVAGE. This figure clearly shows that our tool usually produces fewer but longer contigs.

3.5 Benchmark on simulated biased HIV datasets

A major challenge in viral quasispecies assembly is to reconstruct the low-abundance haplotypes. To evaluate the performance of our methods on assembling low abundance strains, we used HIV strains HXB2 and NL43 to simulate three groups of datasets with extremely biased coverages. We chose HXB2 and NL43 because they share the highest similarity and longest common region among five HIV strains, representing the hardest case for assembly. The total coverage for each group is 1000×, with HXB2-900×, NL43-100×; HXB2-950×, NL43-50×; and HXB2-990×, NL43-10× for each group, respectively. These datasets contain 250 bp paired-end reads produced by ART-illumina with average insert size of 600 bp and standard deviation of 150 bp.

With the similar pre-processing procedures on HIV 5 strains data, we used PEHaplo to assemble contigs from these datasets and compared the results with SAVAGE, which has better performance than MLEHaplo and IVA. The results are shown in Table 3.

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 one long contig. In addition, when the rare strain reached 10% (100×) of the total coverage, PEHaplo could partially assemble it, while SAVAGE could only assemble the dominant one.

3.6 Benchmark on influenza dataset

In addition to HIV data, we also applied PEHaplo on a real Influenza H1N1 dataset (SRR1766219) sequenced from the mix of a wild type (99%) and a mutant type (1%). This dataset is sequenced with Illumina MiSeq sequencing technology, containing

Table 3. Assembly results on simulated biased HXB2-NL43 MiSeq dataset for SAVAGE and PEHaplo

HXB2:NL43	Tools	Contig num	N50	Genome covered (%)	Unaligned length (bp)	Mismatch rate (%)	Indels (%)
900:100	SAVAGE	7	2 500	46.76	581	0.022	0
	PEHaplo	11	8 163	80.26	0	0.038	0
950:50	SAVAGE	8	8 032	46.76	1817	0	0
	PEHaplo	1	9 470	46.76	0	0.033	0.01
990:10	SAVAGE	13	2 130	46.75	1590	0.022	0
	PEHaplo	1	9 509	48.95	0	0	0

Table 4. Assembly results on Influenza MiSeq dataset for SAVAGE and PEHaplo

Tools	Contig num	N50	Genomes covered (%)	Unaligned length (bp)	Mismatch rate (%)	Indels (%)
SAVAGE	220	620	96.3	38 303	0.818	0.046
PEHaplo	10	1 790	99.5	1 270	0.836	0.007

646 879 pairs (2 × 250) of reads covering the two strains to ∼23 000×. The mutant type carries two silent mutations in the M1 ORF (C354T and A645T, segment 7).

We first performed 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. Still, those reads that duplicate at least three times in the raw dataset were kept. After pre-processing, 11 940 reads were kept for strand adjustment and assembly.

It is worth noting that the H1N1 viruses have eight segmented genomes. PEHaplo produced 10 contigs from the MiSeq Influenza data, with eight contigs covering over 99% of the eight segments of Influenza genome and two contigs unaligned. On the other hand, SAVAGE produced 220 contigs with a N50 value of 620 bp. The results are summarized and compared in Table 4. The comparison shows that PEHaplo works much better than SAVAGE on this Influenza quasispecies data as it successfully assembled all the eight segments.

This dataset contains a wild type and a rare mutant type (1%). However, neither method can recover the two mutations in the rare haplotype. In order to investigate this issue, we mapped all reads back to the region of the rare haplotype that contains the two mutations. The read mapping results clearly show that only several reads contain the same bases as the mutant type, while all the other reads support the wild type. Thus, with such low number of mapped reads, existing information is not sufficient to distinguish true mutations from sequencing errors. Long read sequencing platforms might be a better choice for recovering the rare mutant type.

3.7 Computational time and memory usage

To evaluate the computational efficiency of our tool, we compare the running time and peak memory usage of the tested tools on the HIV 5-strain simulated data and also the real data. The results are shown in Table 5. PEHaplo runs significantly faster than SAVAGE and MLEHaplo. All the experiments were tested on a 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 IVA, 16 threads for SAVAGE and 1 thread for PEHaplo. The commands of running

Table 5. Running time and peak memory usage of assembly tools on HIV simulated and real data

Tools	Simulated data		Real data	
	Time	Memory (GB)	Time	Memory (GB)
IVA	17m	2.6	17m	2.6
MLEHaplo	10h	6.4	4h	2.4
SAVAGE	6h	1.5	125h	1.1
PEHaplo	9m	2.9	23m	1.3

these tools on HIV simulated data can be found in Supplementary Material Section S8.

4 Discussion and conclusion

For paired-end reads, one may consider to combining read pairs into a longer sequence before assembly. We applied existing read joining tools for this purpose. However, joining reads is not a trivial task 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 lose the paired-end information for guiding the path finding process. As a result, the experimental results using PEAR (Zhang *et al.*, 2014) and other end merging tools show inferior performance. Therefore, we did not include this 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 viral haplotypes reconstruction are discussed in BAsE-Seq (Hong *et al.*, 2014). With the increased read quality, we expect to see more promising applications of the third-generation sequencing to viral haplotype reconstruction.

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 present PEHaplo: a *de novo* viral haplotype reconstruction tool for viral quasispecies. It does not need references. When the references are available, it may be preferable to use reference-based methods.

Acknowledgements

We would like to acknowledge the help from N.D. on producing some figures and the help from M.V. on proofreading the manuscript.

Funding

This work has been supported by MSU and NSF CAREER [grant number DBI-0953738].

Conflict of Interest: none declared.

References

Allam,A. *et al.* (2015) Karect: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data. *Bioinformatics*, **31**, 3421–3428.

Astrovskaya,I. *et al.* (2011) Inferring viral quasispecies spectra from 454 pyrosequencing reads. *BMC Bioinformatics*, **12**, S1.

Baaijens,J.A. *et al.* (2017) De novo assembly of viral quasispecies using overlap graphs. *Genome Res.*, **27**, 835–848.

Barbosa,V.C. *et al.* (2012) Quasispecies dynamics with network constraints. *J. Theor. Biol.*, **312**, 114–119.

Beerenwinkel,N. *et al.* (2012) Challenges and opportunities in estimating viral genetic diversity from next-generation sequencing data. *Front. Microbiol.*, **3**, 329

Boisvert,S. *et al.* (2012) Ray Meta: scalable de novo metagenome assembly and profiling. *Genome Biol.*, **13**, R122.

Cormen,T.H. (2009) *Introduction to Algorithms*. MIT press, Cambridge, MA.

Di Giallonardo,F. *et al.* (2014) Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. *Nucleic Acids Res.*, **42**, e115–e115.

Domingo-Calap,P. *et al.* (2016) Mechanisms of viral mutation. *Cell. Mol. Life Sci.*, **73**:4433–4448.

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

Haj Rachid,M. and Malluhi,Q. (2015) A practical and scalable tool to find overlaps between sequences. *BioMed . Res. Int.*, **2015**, 1.

Hong,L.Z. *et al.* (2014) BAsE-Seq: a method for obtaining long viral haplotypes from short sequence reads. *Genome Biol.*, **15**, 517.

Huang,A. *et al.* (2011) QColors: an algorithm for conservative viral quasispecies reconstruction from short and non-contiguous next generation sequencing reads. *In Silico Biol.*, **11**, 193–201.

Huang,W. *et al.* (2012) ART: a next-generation sequencing read simulator. *Bioinformatics*, **28**, 593–594.

Hunt,M. *et al.* (2015) IVA: accurate de novo assembly of RNA virus genomes. *Bioinformatics*, **31**, 2374–2376.

Jayasundara,D. *et al.* (2015) ViQuaS: an improved reconstruction pipeline for viral quasispecies spectra generated by next-generation sequencing. *Bioinformatics*, **31**, 886–896.

Jiang,H. and Wong,W.H. (2009) Statistical inferences for isoform expression in RNA-Seq. *Bioinformatics*, **25**, 1026–1032.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**, 357.

Laserson,J. *et al.* (2011) Genovo: de novo assembly for metagenomes. *J. Comput. Biol.*, **18**, 429–443.

Li,M. *et al.* (2017). PECC: Correcting Contigs Based on Paired-End Read Distribution. *Comput. Biol. Chem.*, **69**, 178–184.

Li,Z. *et al.* (2012) Comparison of the two major classes of assembly algorithms: overlap–layout–consensus and de-bruijn-graph. *Brief. Funct. Genomics*, **11**, 25–37.

Malhotra,R. *et al.* (2013) Estimating viral haplotypes in a population using k-mer counting. In: *IAPR International Conference on Pattern Recognition in Bioinformatics*, pp. 265–276. Springer.

Malhotra,R. *et al.* (2015) Maximum likelihood de novo reconstruction of viral populations using paired end sequencing data. *arXiv preprint arXiv: 1502.04239*.

Mangul,S. *et al.* (2014) Accurate viral population assembly from ultra-deep sequencing data. *Bioinformatics*, **30**, i329–i337.

Mikheenko,A. *et al.* (2015) MetaQUAST: evaluation of metagenome assemblies. *Bioinformatics*, **32**, 1088–1090.

Namiki,T. *et al.* (2012) MetaVelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads. *Nucleic Acids Res.*, **40**, e155.

Nowak,M.A. (2006). *Evolutionary Dynamics*. Harvard University Press, Cambridge, MA.

Peng,Y. *et al.* (2011) Meta-IDBA: a de novo assembler for metagenomic data. *Bioinformatics*, **27**, i94–i101.

Peng,Y. *et al.* (2012) IDBA-UD: a de novo assembler for single-cell and metagenomic sequencing data with highly uneven depth. *Bioinformatics*, **28**, 1420–1428.

Prabhakaran,S. *et al.* (2010) HIV-haplotype inference using a constraint-based Dirichlet process mixture model. In: *Machine Learning in Computational Biology (MLCB) NIPS Workshop*, pp. 1–4.

Prabhakaran,S. *et al*. (2014) HIV haplotype inference using a propagating Dirichlet process mixture model. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **11**, 182–191.

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

Salzberg,S.L. *et al*. (2008) Gene-boosted assembly of a novel bacterial genome from very short reads. *PLOS Comput. Biol*., **4**, e1000186.

Schirmer,M. *et al*. (2014) Benchmarking of viral haplotype reconstruction programmes: an overview of the capacities and limitations of currently available programmes. *Brief. Bioinformatics*, **15**, 431–442.

T O'neil,S. and Emrich,S.J. (2012) Haplotype and minimum-chimerism consensus determination using short sequence data. *BMC Genomics*, **13**, S4.

Töpfer,A. *et al*. (2013) Probabilistic inference of viral quasispecies subject to recombination. *J. Comput. Biol*., **20**, 113–123.

Töpfer,A. *et al*. (2014) Viral quasispecies assembly via maximal clique enumeration. *PLoS Comput. Biol*., **10**, e1003515.

Treangen,T. *et al*. (2013) MetAMOS: a modular and open source metagenomic assembly and analysis pipeline. *Genome Biol*., **14**, R2.

Yuan,C. *et al*. (2015) Reconstructing 16s rRNA genes in metagenomic data. *Bioinformatics*, **31**, i35–i43.

Zagordi,O. *et al*. (2011) ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC Bioinformatics*, **12**, 119.

Zhang,J. *et al*. (2014) PEAR: a fast and accurate illumina paired-end read merger. *Bioinformatics*, **30**, 614–620.