De novo assembly of viral quasispecies using overlap graphs

Jasmijn A. Baaijens[1], Amal Zine El Aabidine[2], Eric Rivals[2,3*,†], Alexander Schönhuth[1,*,†]

[1] Centrum Wiskunde & Informatica, Amsterdam, Netherlands
[2] LIRMM, CNRS and Université de Montpellier, Montpellier, France
[3] Institut Biologie Computationnelle, CNRS and Université de Montpellier
[*] These authors contributed equally
[†] Corresponding authors rivals@lirmm.fr, alexander.schoenhuth@cwi.nl

Keywords: de novo assembly, virus, method, high throughput sequencing, HIV, Zika

Abstract

A viral quasispecies, the ensemble of viral strains populating an infected person, can be highly diverse. For optimal assessment of virulence, pathogenesis and therapy selection, determining the haplotypes of the individual strains can play a key role. As many viruses are subject to high mutation and recombination rates, high-quality reference genomes are often not available at the time of a new disease outbreak. We present SAVAGE, a computational tool for reconstructing individual haplotypes of intra-host virus strains without the need for a high-quality reference genome. SAVAGE makes use of either FM-index based data structures or ad-hoc consensus reference sequence for constructing overlap graphs from patient sample data. In this overlap graph, nodes represent reads and/or contigs, while edges reflect that two reads/contigs, based on sound statistical considerations, represent identical haplotypic sequence. Following an iterative scheme, a new overlap assembly algorithm that is based on the enumeration of statistically well-calibrated groups of reads/contigs then efficiently reconstructs the individual haplotypes from this overlap graph. In benchmark experiments on simulated and on real deep coverage data, SAVAGE drastically outperforms the only de novo viral quasispecies assembler available so far. When run on ad-hoc consensus reference sequence, SAVAGE performs very favorably in comparison with state-of-the-art reference genome guided tools. We also apply SAVAGE on a deep coverage (30 000x) sample of a patient infected by the Zika virus, which sheds light on the genetic structure of the respective viral quasispecies.

Introduction

Viruses such as HIV, the Zika and the Ebola virus, populate their hosts as an ensemble of genetically related but different mutant strains, commonly referred to as *viral quasispecies*. These strains, each characterized by its own haplotypic sequence are subject to high mutation and recombination rates (Domingo et al. 2012; Duffy et al. 2008). Sequencing methods aim at capturing the genetic diversity of viral quasispecies present in infected samples; the promise is that next-generation sequencing (NGS) based methods will, as soon as possible, assist clinicians in selecting treatment options and other clinically relevant decisions.

Ideally, a *viral quasispecies assembly* characterizes the genetic diversity of an infection by presenting all of the viral haplotypes, together with their abundance rates. There are two major challenges in this.

(1) The number of different strains is usually unknown. Further complicating factors are that two different strains can differ by only minor amounts of distinguishing mutations in certain regions of the genome and that abundance rates can be as low as the sequencing error rates, which hampers the detection of true low-frequent mutations.

(2) Due to the great diversity and the high mutation rates, reference genomes representing high-quality consensus genome sequence can be obsolete at the time of the outbreak of the disease. Of course, the lack of a suitable reference genome can be a major hindrance.

It is important to understand that all existing assembly methods fail to address either the first or the second point. While several *reference-guided approaches* (see below for related work) specialize in viral quasispecies assembly and soundly address point (1), the vast majority of them depends on high-quality reference sequence as a backbone to their approaches. Hence, when confronted with hitherto unknown, significantly deviating mutation patterns, these approaches fail to perform sufficiently well. On the other hand, *de novo assembly approaches* (see again below for related work) do not depend on reference genomes. These approaches however aim at assembling consensus genomes rather than strain-specific sequences. A further issue is that nearly all of the NGS based genome assemblers rely on the *de Bruijn graph* as assembly paradigm. Thereby, reads are decomposed into k-mers, where k is usually considerably smaller than the read length. As above mentioned, it is imperative in viral quasispecies assembly to distinguish low-frequency mutations from sequencing errors. While lowly frequent mutations are genetically linked, hence co-occur within different reads, sequencing errors do not exhibit patterns of co-occurrence. The detection of patterns of co-occurrence is decisively supported by examining reads at their full length. However, only rarely, NGS de novo assembly approaches are based on overlap graphs, which make use of full-length reads and do not decompose them into smaller parts.

There are two exit strategies to this twofold dilemma behind the possible lack of a reference genome. To the best of our knowledge, none of them have been sucessfully explored so far; we would like to explore both of them here.

The *first strategy* is to construct an overlap graph directly from the patient sample reads. Subsequently, one employs a ploidy-aware assembly algorithm that can extract strain-specific sequences from overlap graphs. The challenge is that constructing overlap graphs requires a pairwise comparison of all reads, which, for deep coverage data sets, requires sophisticated indexing techniques to be feasible. Here, we show how to make efficient use of FM-index based techniques (Välimäki et al. 2012) to construct overlap graphs without any need for a reference genome. As such, we provide *the first approach for de novo assembly of viral quasispecies based on overlap graphs*.

The *second strategy* is to construct consensus genome sequence from scratch (a.k.a. the patient samples themselves), using one of the available de novo consensus genome assemblers (among which, the most popular tool is VICUNA (Yang et al. 2012)), and to subsequently run one of the reference-guided approaches using this ad-hoc consensus as a reference. Rather surprisingly, we found none of the extant state-of-the-art reference-guided approaches (Zagordi et al. 2011; Prabhakaran et al. 2014) to work sufficiently well under these circumstances. However, the algorithm used in HaploClique (Töpfer et al. 2014) provided a source of

inspiration. This method is also reference guided, but uses the reference solely for providing anchor points for constructing an overlap graph (Töpfer et al. 2014). Unlike in many other approaches, the exact sequence of the haplotypes is then assembled from the reads, and not from the reference.

The fact that HaploClique, as the only approach providing inspiration for the latter idea, is overlap graph based unifies the two apparently different looking strategies: for both of them the construction of an overlap graph is required in the first place. After construction, there is no need for a particular choice of an overlap graph based assembly algorithm—any good such algorithm will apply in both cases.

While providing inspiration in general, the overlap graph assembly algorithm presented in (Töpfer et al. 2014) has proven too expensive—already data sets of about 1000x coverage require excessive computational resources. The reason is that it is based on the enumeration of maximal cliques, which is exponential in the read coverage, both in terms of runtime and space. We therefore present a novel algorithm for this part of overlap graph based viral quasispecies assembly. We show that this algorithm, while being similar in spirit to enumerating maximal cliques, is two orders of magnitude faster and also decisively more efficient in terms of space consumption. At the same time, this algorithm achieves performance rates in terms of common assembly quality measures (Mikheenko et al. 2016, MetaQUAST) near-identical to maximal clique enumeration and superior to all other reference-guided approaches on ad-hoc consensus reference.

In summary, we make relevant contributions for

(i) the construction of overlap graphs from deep coverage read data and

(ii) viral quasispecies assembly using the overlap graph assembly paradigm.

In combination, we present SAVAGE (Strain Aware VirAl GEnome assembly), a method that allows for reference-free assembly of viral quasispecies from sequencing data sets of truly deep coverage. In this, we do not only provide the first genuine de novo viral quasispecies assembly approach based on overlap graphs, but we also provide the first method that can exploit ad-hoc consensus sequence generated from patient samples, as computed by (Yang et al. 2012, VICUNA) for example, for high-performance viral quasispecies assembly.

Related Works Several recent *reference-guided approaches* suggested statistical frameworks modeling the driving forces underlying the evolution of viral quasispecies. While previous approaches focused mostly on local reconstruction of haplotypes (Huang et al. 2012; Quince et al. 2011; Zagordi et al. 2011; 2010), more advanced approaches aimed at global reconstruction of haplotypes, for example, by making use of Dirichlet process mixture models (Prabhakaran et al. 2014), hidden Markov models (Töpfer et al. 2013), or sampling schemes (Prosperi and Salemi 2012). There are also recent combinatorial approaches which computed paths in overlap graphs (Astrovskaya et al. 2011), enumerated maximal cliques in overlap graphs (Töpfer et al. 2014), and computed maximal independent sets in conflict graphs (Mangul et al. 2014).

Numerous *de novo assembly approaches* aim at reconstructing a consensus genome sequence from NGS data – (Bankevich et al. 2012; Gnerre et al. 2011; Luo et al. 2012; Simpson et al. 2009; Zerbino and Birney 2008) to list a few. For a clear exposure of their merits, we refer the readers to papers presenting comparative evaluations (Bradnam et al. 2013; Gurevich et al. 2013; Salzberg et al. 2011) to references therein. The only de novo assembly approach for NGS data based on a derivative of overlap graphs is SGA (Simpson and Durbin 2012), which efficiently builds a string graph using a FM-index of the reads (Ferragina and Manzini 2000), certainly a source of inspiration for our work. Last but not least, of course, the spectrum of methods include de novo assemblers dedicated for computing a single consensus sequence for viral quasispecies patient samples (Hunt et al. 2015; Yang et al. 2012). Since VICUNA (Yang et al. 2012) is the most popular tool in this category, we will use it to get a reference sequence that serves as starting point for reference based methods.

There are a few exceptions from the above cited work. MLEHaplo (Malhotra et al. 2016b) is, to the best of our knowldege, the only existing de novo approach for haplotype-resolved viral quasispecies assembly.

MLEHaplo is based on a de Bruijn graph, and we indeed found it to suffer from certain limitations in our evaluation. VGA (Mangul et al. 2014) addresses to make use of ad-hoc consensus sequence for viral quasispecies assembly. There are also de novo *metagenome assembly* approaches (Laserson et al. 2011; Nurk et al. 2016) which expect collections of genomes from different species as input, and *gene assembly* methods (Gregor et al. 2016; Zhang et al. 2014b), which operate reference-free by aligning reads to gene template sequence as provided by suitable databases.

Results

We have designed and implemented SAVAGE (Strain Aware VirAl GEnome assembly), a method for *de novo* viral quasispecies assembly based on overlap graphs. In this section, we provide a high-level description of the algorithmic approach and analyze its performance, also in comparison to state-of-the-art approaches. Finally, we present assembly results using SAVAGE on a sample from a patient infected by the Zika virus. We refer to the Methods section for any methodical details.

Approach

Our algorithm proceeds in three stages (panel A of Figure 1), each of which iteratively clusters the input sequences and extends them to unique haplotypes. While *Stage a* has the original reads as input and contigs as output, *Stage b* has these contigs as input and maximally extended contigs as output. Contigs are supposed to reflect individual haplotype sequences. Finally, the optional *Stage c* merges maximized contigs into master strain sequences. This reflects the existence of master strains in many viruses, where each individual haplotype deviates from one of the master strains by only a relatively minor amount of mutations. Each stage is divided into **overlap graph construction** (upper part of panel C in Figure 1) and **overlap graph based assembly** (lower part of panel C in Figure 1). Between the stages, this generic structure only differs in the details.

The strength of overlap graphs for viral quasispecies assembly is that we can distinguish sequencing errors from true mutations by posing very strong constraints on the overlaps in terms of minimal overlap length and of sequence similarity. In addition, we also employ paired-end read information. This results in a very conservative overlap graph, where an edge indicates that two sequences are very likely to originate from the same virus strain. Therefore, by enumerating cliques in the overlap graph we cluster the reads per strain, thus reconstructing the individual haplotypes of the viral quasispecies.

We construct overlap graphs in two steps: first, pairs of reads are determined that share sufficiently long and well-matching overlaps, followed by a statistical evaluation of the quality of each of the overlaps. We explore two options for finding all such overlap candidates. The first option is to apply a completely *de novo* procedure using FM-index based techniques (Välimäki et al. 2012). The second option is to align all reads against a reference genome, such that read-to-read alignments can be induced from the read-to-reference alignments. However, in case of a viral outbreak there may not be a suitable reference genome available; we target such cases by constructing an ad-hoc consensus sequence from the patient samples, as computed by (Yang et al. 2012, VICUNA).

SAVAGE offers three different modes, corresponding to the different approaches to overlap graph construction described above: **SAVAGE-de-novo** uses the first option and is therefore completely reference-free, while **SAVAGE-ref-bg** uses the second option and thus relies on a bootstrap reference sequence. For benchmarking purposes we also consider **SAVAGE-ref-hg**, which takes as input a high quality reference sequence.

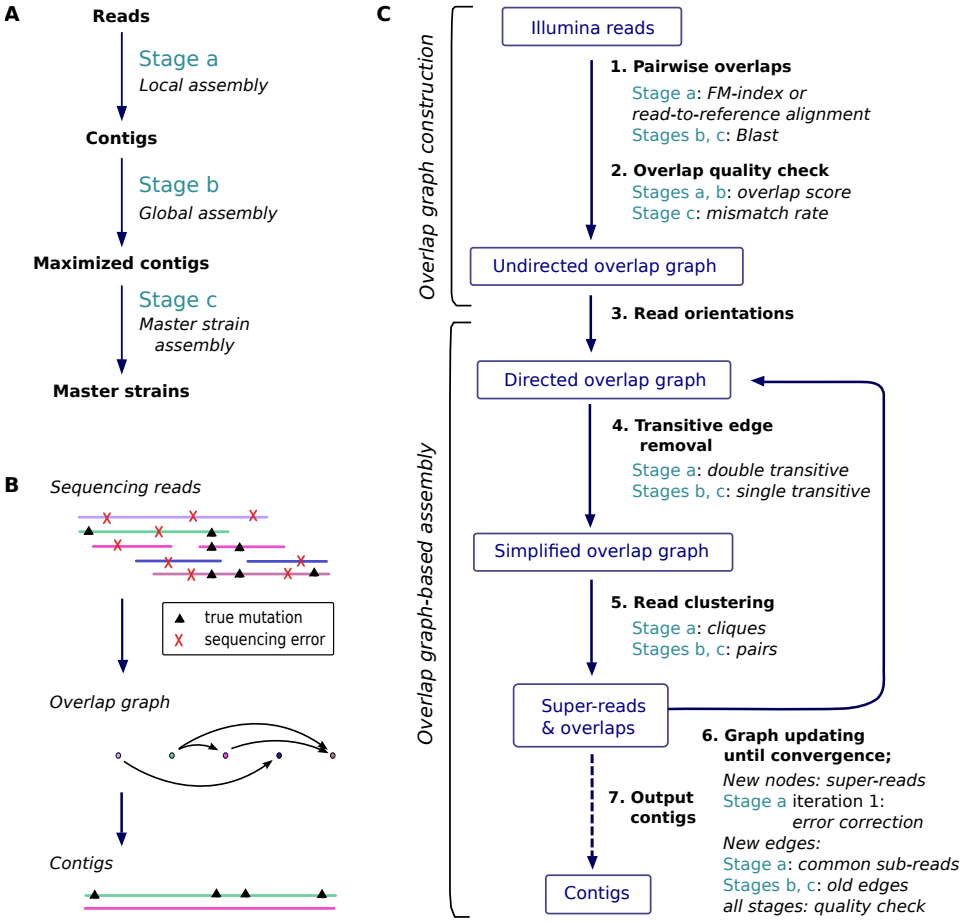

Figure 1: An overview of the workflow and algorithms of SAVAGE. **A.** The three stages of SAVAGE. Each assembles sequences into longer sequences. For clarity, we assign different names to the sequences output by each stage: contigs, maximized contigs, and master strains, respectively. **B.** Principle of overlap graph construction and distinction among the reads between errors and shared mutations. **C.** Each stage has two steps: first, the overlap graph construction, second, assembly. This panel summarizes the differences in each step between the three stages.

Benchmarking data

For benchmarking experiments and performance analysis, we considered several data sets.

Simulated mix. We created a simulated data set from a mixture of five HIV-1 strains (HXB2, JRCSF, 89.6, NL43, and YU2). These strains are of length 9478–9719 bp and pairwise sequence identity varies between 93.6–98.6%. Our first benchmarking data set, referred to as the *simulated mix*, consists of simulated Illumina MiSeq reads (2x250 bp) from a homogeneous mixture with total coverage of 600x (i.e., 120x per strain), obtained using the simulation software SimSeq (see Methods).

Lab mix. Our second benchmarking data set consists of real Illumina MiSeq reads (2x250 bp) with 20000x coverage, obtained from a lab mix of the same five HIV-1 strains as described above, which was recently presented as a gold standard benchmark (Di Giallonardo et al. 2014). We will refer to this data set as the *lab mix*.

Divergence-vs-ratio. To analyze the combined effect of the levels of divergence and of their relative abundance of the strains, we constructed 36 additional data sets as follows. Starting from the HIV-1 89.6 haplotype, we created six alternative haplotypes by introducing, respectively, 0.5%, 0.75%, 1%, 2.5%, 5%, and 10% random mutations. For each of those six alternative strains, we created six data sets by simulating reads (2x250 bp Illumina MiSeq) from the mutated strain and the original at a ratio of 1:1, 1:2, 1:5, 1:10, 1:50, and 1:100, respectively, with a total coverage of 500x per data set.

Zika virus sample. We applied SAVAGE to a sample of Asian-lineage Zika virus (ZIKV) consisting of Illumina MiSeq 2x300 bp sequencing reads (∼30,000x coverage) obtained from a rhesus macaque after four days of infection (Dudley et al. 2016, animal 393422).

Evaluation preliminaries

In the case of a viral outbreak, the agent and its genome may be unknown (or may have significantly diverged from closely related strains such that available reference sequences are potentially inadequate for analysis), and the samples taken from infected patients contain an unknown number of divergent strains. Here, we target such cases where no reference genome is available. Of course, a sample sequenced with Next Generation Sequencing delivers enough reads and sufficient coverage to allow a *de novo* assembly of a viral genome (here, we mean a single genome assembly, not a quasispecies assembly). However, such genome sequences may not represent well any of the true viral haplotypes present in the sample.

In the sequel, all assembly algorithms were run using default settings. Evaluations of assemblies were performed with MetaQUAST (Mikheenko et al. 2016), which computes the usual statistics – number of contigs, largest contigs, N50, misassembled contig length, target genome(s) covered, and error rates – and we accounted only for contigs larger than a threshold of 500 bp. A contig is called misassembled if it contains at least one misassembly, i.e., a position where the left and right flanking sequences align to the true genomes with a gap or overlap of more than 1 kbp, or align to different strands, or even align to different strains.

Dependence of reference based approaches on reference genome quality

Reference based quasispecies assembly tools proved to perform adequately when a high quality reference genome is available (Zagordi et al. 2011; Prabhakaran et al. 2014). We question whether reference based approaches could generate appropriate quasispecies assemblies if provided with a *de novo* assembled genome sequence obtained from the sample reads, rather than a high quality reference genome. In other words, can one bootstrap a reference based quasispecies assembly with a reference genome assembled from the sample reads? To address this question, we compared state-of-the-art methods PredictHaplo (Prabhakaran et al. 2014) and ShoRAH (Zagordi et al. 2011) on both simulated and real HIV-1 datasets in two conditions: either with the high quality HIV-1 reference genome, or with a genome sequence obtained by running the

VICUNA assembler on the sample reads. For simplicity, we distinguish both cases, we term the former **a high quality reference genome** (denoted hg; here the HIV-1 reference genome) and the latter a **bootstrap reference genome** (denoted bg). The quality of the output assemblies, as evaluated with MetaQUAST, are described in Table 1.

PredictHaplo and ShoRAH yield a set of sequences, each of which is supposed to be a strain present in the sample. Each output sequence is somehow derived by altering the reference sequence; this is why all of their output sequences have the same length. In comparison, MLEHaplo (Malhotra et al. 2016b) and SAVAGE proceed by assembling reads and output sequences that are of different lengths (hence, the N50 provides an indication of the contig length distribution).

PredictHaplo yields 4 or 3 sequences which logically cover 75% and 60% of the five strains of the mix. However, when using VICUNA reference, PredictHaplo produces sequences that are 100% misassembled. So providing it with a bootstrap reference hinders PredictHaplo to correctly assemble the strains. The same situation occurs with real data (*lab mix*): it provides exactly 5 sequences, thus covering nearly 100% of the expected five strains, but again all sequences are misassembled when using a bootstrap reference.

ShoRAH outputs a much larger number of strains (39/63) on simulated data, (160/169) on real data, which induces a high target genome coverage. When ShoRAH is given the bootstrap reference, on simulated data, 1.6% of these strains are misassembled, while with real data 89% are misassembled. At least on real data, ShoRAH suffers from the same failure as PredictHaplo: the quality of their assembly is highly dependent on that of the reference genome sequence. PredictHaplo and ShoRAH are valuable tools when the reference genome is closely related to sample strains, but seem inadequate to handle cases where a good reference genome is unavailable. This emphasizes the need for new assembly approaches that are independent of a reference genome.

SAVAGE evaluation

For the sake of comparison, we ran SAVAGE on the same HIV-1 benchmarks as above (*simulated mix* and *lab mix*) in both *de novo* mode and reference mode, both with default parameters. The *lab mix* of 20 000x coverage was split into patches of 500x each, on which we applied SAVAGE *Stage a* (Supplemental Figure S1). Subsequently, all *Stage a* contigs were unified and used as input for *Stage b*. Table 1 presents the evaluation results of the *Stage b* maximized contigs for each of the three modes: SAVAGE-ref-hg with HIV-1 genome as reference, SAVAGE-ref-bg with the genome assembled by VICUNA, and SAVAGE-de-novo (without reference). For completeness, we also ran MLEHaplo, another *de novo* approach (Malhotra et al. 2016b). Unfortunately, MLEHaplo could only process the simulated data, but not the real HIV-1 benchmark. Remember that SAVAGE and MLEHaplo proceed by progressively assembling longer and longer contigs starting from the raw reads, until finally, each output contig may cover partially the target genomes. Hence, unlike for PredictHaplo and ShoRAH, the number of contigs shall not be interpreted directly as a number of strains.

On the *simulated mix*, MLEHaplo outputs a set of contigs that cover 56% of the five true genomes and its largest contig is almost equal to the expected genome length ($\approx$ 10 Kbp). However, the overall mismatch rate equals 3.4%, greatly exceeding the error rate of the sequencing reads (1%). With a reference, the results of SAVAGE-ref-hg and SAVAGE-ref-bg, are very similar: the contigs cover from 94% to 99 % of the five genome strains, with an N50 above 5000 bp, and largest contigs above 8.5 Kbp. The mismatch, indel, and N rates are globally better than those offered by PredictHaplo and ShoRAH: the indel and N rates are respectively one or two orders of magnitude lower. Above all, the contigs are free of misassemblies (MAC length is 0%). Strikingly, providing a high quality reference genome or a bootstrap genome makes little difference, and on some statistics SAVAGE with a bootstrap genome achieves better results (less contigs, largest longest contig, higher N50, lower mismatch rate). These observations on the comparison of SAVAGE-ref-hg and SAVAGE-ref-bg are also true on the real HIV-1 benchmark, where the results are even closer. On simulated

	# contigs $\geq$ 500bp	largest contig	N50	MAC length (%)	genomes covered (%)	N-rate (%)	mismatches (%)	indels (%)
Simulated mix								
PredictHaplo								
HIV-1 ref	4	9710	9710	0	79.5	0.659	1.374	0.091
VICUNA ref	3	9800	9800	100	59.1	0.289	1.628	0.175
ShoRAH								
HIV-1 ref	39	9526	9526	0	98.0	0.318	0.394	0.087
VICUNA ref	63	9657	9657	1.6	97.2	0.307	1.356	0.151
MLEHaplo	185	9104	6960	0	56.0	0	3.396	0.078
SAVAGE								
de novo	23	9626	5369	0	90.5	0.001	0.724	0.018
HIV-1 ref	23	8517	5148	0	99.0	0.003	0.402	0.004
VICUNA ref	19	9470	6315	0	94.2	0.002	0.270	0.002
Lab mix								
PredictHaplo								
HIV-1 ref	5	9642	9642	0	99.2	0.259	0.615	0.104
VICUNA ref	5	11000	11000	100	94.5	0.425	0.011	0.136
ShoRAH								
HIV-1 ref	160	9581	9581	0	98.9	0.378	3.203	0.113
VICUNA ref	169	10854	10854	89.3	99.0	0.770	0.911	0.165
SAVAGE								
de novo	482	4256	1062	0	90.5	0.015	0.147	0.048
HIV-1 ref	1003	2560	786	0.3	93.2	0.198	0.182	0.040
VICUNA ref	901	2533	775	0	93.0	0.125	0.283	0.047

Table 1: Assembly results per method (PredictHaplo, ShoRAH, MLEHaplo, and SAVAGE), only taking contigs of at least 500bp into account. The upper part of the table presents results on a collection of simulated Illumina reads from a mixture of five HIV-1 strains. The lower part shows similar results for a real Illumina MiSeq data set, obtained from a mixture of the same five HIV-1 strains. Unfortunately, MLEHaplo failed to run on this data set.
The *N50* score gives the maximal length for which the collection of all contigs of that length or longer covers at least half of the total assembly length. The *MAC length* statistic reflects the proportion of misassembled contig length compared to the total assembly length. After aligning the contigs to the true haplotypes, we obtain the fraction of the *target genomes* that is covered by the contigs. Finally, we give the *N-rate*, which is the fraction of uncalled bases in the assembly, along with the *mismatch and indel rates*. For these measures an uncalled base does not count as a mismatch and the length of an indel is not taken into account.

data, SAVAGE-de-novo delivers an assembly that is qualitatively equivalent to the SAVAGE-ref-hg and -bg assemblies: its coverage reaches only 90% but it has the longest contig of all, and still no misassemblies. On real HIV-1 data, all SAVAGE modes yield assemblies that are more fragmented (larger number of contigs, smaller N50) than for simulated data. However, SAVAGE-de-novo performs better than the versions with a reference: it yields less contigs, a larger N50 value, no misassemblies, and lower N and mismatch rates. Compared to PredictHaplo and ShoRAH, SAVAGE-de-novo offers contigs with improved mismatch and indel rates, hence of higher sequence quality.

Overall, SAVAGE can process samples containing a mixture of 5 strains and recover most of the target genomes with high level of sequence quality. It performs slightly better in *de novo* mode than with a reference sequence on real data. Moreover, compared to existing methods, it does not suffer from misassemblies and it can take advantage of a bootstrap reference sequence built by an single genome assembler.

In terms of CPU time, SAVAGE-ref-bg was faster than SAVAGE-de-novo, with 24 versus 91 minutes on the *simulated mix* and 171 versus 1331 minutes on the *lab mix* (Supplemental Table S2). This was to be expected, since *de novo* overlap graph construction requires enumeration of all approximate suffix-prefix overlaps among the reads. In comparison, PredictHaplo was faster on the simulated data (7 min) but comparable to SAVAGE-de-novo on the real data (158 min). ShoRAH was also faster on *simulated mix* (12 min) but very slow on the *lab mix* (22256 min). Finally, with 54 minutes MLEHaplo was in between SAVAGE-de-novo and SAVAGE-ref-bg. Peak memory usage varied between 0.04 GB (PredictHaplo) and 2.2 GB (MLEHaplo) for the simulated data, and between 0.5 GB (SAVAGE-de-novo/SAVAGE-ref-bg) and 12 GB (ShoRAH) for the real data. A complete comparison is presented in Supplemental Table S2.

Effect of strain divergence and relative abundance

Assembling the sequences of several master strains from a viral sample may turn out more difficult depending on both the level of strain divergence and on their relative abundance. After comparing SAVAGE to state-of-the-art methods, we investigate the ranges of divergence levels and of relative abundance that SAVAGE can properly handle, and examine the combined effect of these two parameters on the assembly quality. We use a series of 36 benchmark datasets simulated from two HIV-1 strains: a combination of six divergence levels (from .5 % until 10% of nucleotidic divergence) with six ratios of abundance (from 1:1 until 1:100). We run SAVAGE-de-novo and SAVAGE-ref-bg (i.e., with VICUNA assembled genome). All assemblies were evaluated with MetaQUAST, and Figure 2 reports the heatmaps of (A) the coverage fraction of the two genomes, (B) the mismatch rate, and (C) the absolute error on the frequency estimates of each strain.

Comparing the two modes of SAVAGE, *de novo* or with a bootstrap reference, we observe similar results and a slight advantage to SAVAGE-de-novo in terms of genomes coverage. Altogether, SAVAGE obtains quasispecies assemblies of good coverage and low mismatch rates for all divergence levels and all relative abundance ratios starting from 1:1 until 1:50, proving its ability to distinguish sequencing errors from true mutations. A very low divergence level of 0.5% hinders SAVAGE-de-novo from distinguishing the two strains for a ratio of 1:10. Similarly, an extreme relative abundance of 1:100 hinders SAVAGE to reconstruct both strains (genome coverage values around 50% meaning that only one of the two strains has been assembled).

Capacity to estimate the frequency of each strain SAVAGE also computes an estimate of each strain frequency in the sample. For these synthetic benchmarks, we compared the estimated frequency of the major strain in the sample with the real frequencies. The rightmost panel of Figure 2 shows the absolute difference between the estimated frequency and the true frequency. This comparison was performed only when the strains were almost fully assembled (exactly two strains of length $\geq$ 5000 bp), hence abundance ratios of 1:50 and 1:100 were excluded. Of the remaining 24 datasets, 4-6 samples did not satisfy these criteria; the

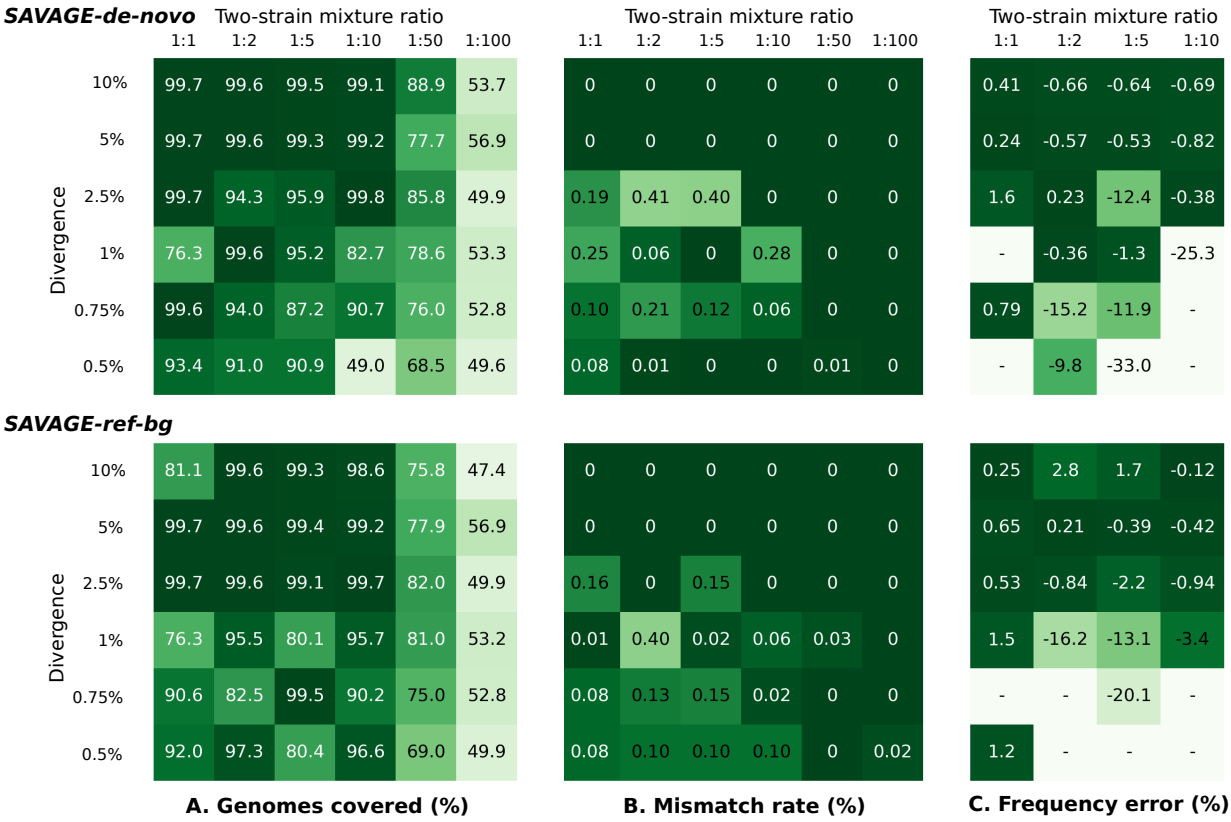

Figure 2: Performance of SAVAGE-de-novo and SAVAGE-ref-bg, depending on pairwise distance and mixture ratio. **A.** Target genome fraction recovered (%) considering all contigs $\geq$ 500 bp. **B.** Overall mismatch rate (%) considering all contigs $\geq$ 500 bp. **C.** Absolute error of estimated frequency for the major strain (%). A positive error indicates overestimation, while a negative error indicates underestimation. Only frequency estimates for assemblies containing exactly two strains longer than 5000 bp were evaluated.

corresponding entries are marked '-' in the heatmaps. In general, the estimated frequencies are close to the real ones provided the divergence level lies strictly above 0.75% (an average difference of less than one point). The estimates are less accurate for smaller divergence levels (up to 33% for SAVAGE-de-novo and 20% for SAVAGE-ref-bg). A majority of negative values indicates a tendency to underestimate the major strain. Hence, SAVAGE provides reasonable frequency estimates of the strains in a wide range of divergence levels (from 1% to 10%), but additional work is needed to explore complex mixtures of more strains.

Zika virus sample

To test SAVAGE-de-novo on real conditions, we ran it on a sample taken from a resus macaque infected by Zika virus (ZIKV) (Dudley et al. 2016). Using a similar procedure as for the real HIV data, we split the reads into patches of 500x each and proceeded with *Stage a* assembly on each patch (Supplemental Figure S1). Subsequently, the whole collection of *Stage a* contigs was used together as input for *Stage b*, which yielded 92 contigs ($\geq$ 500 bp), six of which were longer that 2500 bp. The N50 measure was 1579 bp, the largest contig being 4153 bp long. Aligning the contigs to the established ZIKV reference genome[1] reveals that the 10767 bp reference genome was covered between positions 225–10767, the greatest divergence occurring between positions 1700 and 4200 with circa 20 to 30 different haplotypes. In *Stage c*, we merged the strains of less than 1% divergence, thus reducing the number of output sequences (now called master strains) to 9 (two of which being $\geq$ 2500 bp) and increasing the N50 measure to 5082. The two largest master strains counted 5082 bp and 8311 bp, together covering the ZIKV reference genome between positions 444 and 10387. The resulting virus strains diverge by more than 10% from the reference genome, but only 0.8% from a complete ZIKV genome of the Asian lineage (GenBank sequence KU681081.3). The major difference between the two strains was a one nucleotide deletion at position 4088, followed by two SNPs at positions 4090 and 4093, respectively. Our frequency estimation procedure predicted the haplotype harboring the deletion to be the minor haplotype with a frequency of 38.7%, compared to 61.3% for the major haplotype. We expect that in the future, novel external data obtained by different means will become available for this sample, allowing an in-depth validation of our two-strain quasispecies assembly.

Discussion

Recent outbreaks of viral diseases, such as the Ebola or the Zika virus, have pointed out a pressing need for methods to assess the genetic diversity of viral infections in a flexible manner and without depending strongly on the quality of a reference genome. Here, we have presented SAVAGE, the first method for *de novo assembly of viral quasispecies* based on overlap graphs. Our evaluations have shown that SAVAGE clearly outperforms the only other existing de novo approach for viral quasispecies assembly. SAVAGE can also be run in reference guided mode. In this mode, it performs very favorably in comparison with reference guided tools when using an ad-hoc consensus reference of possibly low quality instead of well curated, high-quality reference sequence. While previous tools tend to become confused by the low quality of the reference, SAVAGE behaves in a robust manner with respect to reference quality. Since reference independent tools are desirable, when confronted with a rapidly spreading outbreak of a highly diverse virus, SAVAGE has proven to bridge a significant gap in the spectrum of viral quasispecies assembly approaches.

We believe that the central reason explaining the benefits of our approach is the use of overlap graphs as the underlying assembly paradigm. While assembling genomes of low ploidy usually works favorably based on de Bruijn graphs, we have pointed out that using reads at their full length obviously is key in assembling viral quasispecies, where distinguishing between low-frequency mutations and sequencing errors is imperative. Using overlap graphs does not only improve the detection of low-frequency strains, but also of true,

[1] `http://www.ncbi.nlm.nih.gov/genome/viruses/variation/Zika/`

and hence co-occurring, mutations. This allows correction of sequencing errors in novel ways, by making integrative use of sound statistical sequence models in combination with an iterative algorithmic scheme for extending reads into contigs of gradually increasing length. In summary, SAVAGE is also amenable to the assembly of low-frequency strain sequences.

Here, key to reference free construction of overlap graphs is the use of FM-index based techniques in a way that is novel in the context of the analysis of viral data. Moreover, we have demonstrated that overlap graphs also seem to be the approach of choice when aiming to make use of ad-hoc consensus reference genomes, such as provided by specialized tools that construct a single consensus sequence from patient sample read data. Often, the resulting consensus sequence is of worse quality than a well-curated reference sequence. This can substantially disturb approaches that rely on the underlying reference as a sequence template (e.g. PredictHaplo, ShoRAH). Overlap graphs constructed by making use of reference sequence coordinates prove to provide a robust alternative, since they use the reference sequence only as a coordinate system for the determination of overlaps.

A few more things are noteworthy. First, the bootstrap reference approach SAVAGE-ref-bg has proven to outperform reference guided approaches in terms of the error rates of the contigs, even when they make use of high-quality reference sequence. On the lab mix benchmark, consisting of real sequencing data, the fully de novo approach SAVAGE-de-novo produced contigs of even higher quality, proving its ability to distinguish sequencing errors from true mutations. Second, although being superior, we have also found our approach to depend on the quality of the reference sequence in case of real sequencing data: for SAVAGE-ref-hg we obtained slightly longer contigs with lower mismatch rates compared to SAVAGE-ref-bg (see Table 1). This, of course, had to be expected: if reference coordinates are too mistaken, overlaps cannot be detected. Together, these two points underline the general benefits of a full de novo viral quasispecies assembly approach.

Of course, there is still room for improvements. Our algorithm, while substantially faster and more space efficient than previous overlap graph based viral quasispecies assembly algorithms, has been particularly tailored towards dividing deep coverage datasets into chunks of 500 to 1000x, and merging the contigs of the chunks in subsequent steps, because this reflects its statistical calibration. While this works perfectly well, it sets certain limits on the frequency of strains we can recover—haplotypes of frequencies below 1% remain difficult to reconstruct. In future work, we will seek to lower these limits further by considering novel strategies for computing cliques in overlap graphs. Last but not least, we will also explore alternative indexing techniques that allow for more relaxed definitions of overlaps and faster computation.

Methods

Overlap graph construction

We first provide a brief definition of an overlap graph and then sketch how to construct such graphs from patient sample read data using *indexes* or *reference genomes* as two options.

Overlap graphs. For a collection $\mathcal{R}$ of sequencing reads (*Stage a*) or contigs (*Stages b,c*), both of which are sequences over the alphabet of nucleotides $\{A, C, G, T, N\}$ (which includes N as a common placeholder for unknown nucleotides), the *overlap graph* $G = (V, E)$ is a directed graph, where vertices $v \in V$ correspond to reads/contigs $R \in \mathcal{R}$ and directed edges connect reads/contigs $R_j, R_j \in \mathcal{R}$ whenever a suffix of R_i of sufficient length matches a prefix of R_j and $\mathrm{QS}(R_i, R_j) \geq \delta$ where $\mathrm{QS} : V \times V \to \mathbb{R}$ is a quality score that has to exceed a certain threshold δ. For *Stages a, b* we make use of the statistical model presented in (Töpfer et al. 2014), where $\mathrm{QS}(R_i, R_j) \geq \delta$ reflects that the overlapping parts of reads R_i and R_j statistically highly likely present a locally identical haplotypic sequence (see Supplemental Methods). In *Stage c*, $\mathrm{QS}(R_i, R_j)$ reflects the fact that the two contigs share only a limited amount of mismatches in their overlaps, meaning that they did likely emerge from identical master strain sequences.

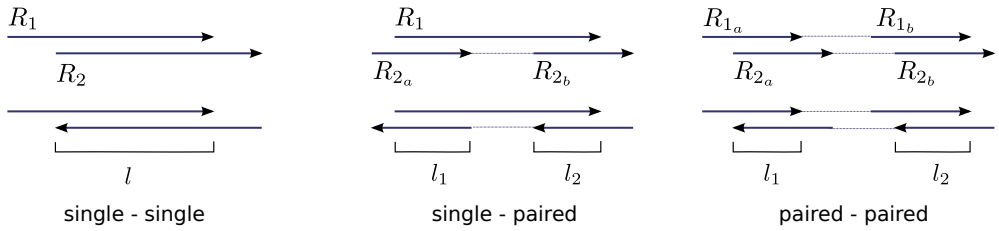

Figure 3: Edge criteria. For an overlap to become an edge in the overlap graph, it must satisfy three criteria. First, the overlap length l must be at least the minimal overlap length L. Second, the overlap quality score $QS(R_1, R_2)$ must be at least the minimal score δ. For overlaps involving paired-end reads, we require both $l_1 \geq L$ and $l_2 \geq L$, and, analogously, $QS(R_{1_a}, R_{2_a}) \geq \delta$ and $QS(R_{1_b}, R_{2_b}) \geq \delta$. Finally, we only accept overlaps where the sequence orientations of a paired-end read agree: either both sequences in forward orientation, or both sequences in reverse orientation.

Paired-end reads. SAVAGE was designed for Illumina MiSeq paired-end reads; after merging self-overlapping pairs, the input in *Stage a* may contain paired-end reads as well as single-end reads. To make use of the pairing information, we add another edge restriction by allowing only the overlap cases shown in Figure 3. For overlaps involving a paired-end read, we require both read ends to have a sufficiently long overlap and a high enough quality score.

Construction. Construction of overlap graphs always proceeds in two steps. First, pairs of reads (R_i, R_j) are determined that have a sufficiently long and well-matching overlap. Subsequently, QS is evaluated on all pairs (R_i, R_j). For *Stages b and c*, where the input is sufficiently small, the first step is implemented by pairwise comparison of all contigs using BLAST (Altschul et al. 1990). The only difficulty is the first step in *Stage a*, where the input is very large (the original deep coverage data). This requires some sophistication; we explore two options:

1. With a read index: We determine all sufficiently long overlaps between sequencing reads using FM-index based techniques (Välimäki et al. 2012, SFO) such that overlaps contain at most 2% mismatches (accounting for 1% sequencing errors in each of the reads). This method, however, only works on single-end reads, so we first ignore the paired-end relations and consider each of the sequences as a single-end read. Then, after listing all pairwise overlaps with SFO, we reconsider the pairing information, outputting only overlaps that are supported by both read ends as described above.

2. With a Reference Genome: We align all reads against a reference genome; here we may use an ad-hoc consensus genome obtained by running an assembly software on the sample reads. With all read alignments in hands, it is then computationally straightforward to determine all sufficiently long and sufficiently matching overlaps pairs.

Read orientations. When merging multiple reads into one consensus sequence, it is important that the reads agree on their respective orientations. Therefore, we apply a read orientation routine that assigns a label $(+/-)$ to every read, indicating the orientation in which its sequence should be considered. This routine starts by setting the orientation of a node of minimal in-degree to $+$, then recursively labels all out-neighbors as defined by the corresponding edges (Figure 4, panel A). When there is no perfect labeling possible, meaning that there are conflicts among the read orientations due to inversions, we heuristically search for an orientation that leads to a minimal amount of conflicts among the reads.

Overlap graph based assembly

In all *stages*, our algorithm proceeds as an iterative procedure where contigs grow with the iterations. The final contigs (in particular the output of *Stage b*, or, optionally, *Stage c*) can substantially exceed the length

of the original reads. As our analyses demonstrate, these contigs present haplotype specific sequences with high accuracy.

Cliques and super-reads. The main idea of our algorithm is to compute cliques in the overlap graph. A *clique* is a subset of the nodes such that each pair of nodes is linked by an edge. By definition of the edges, a clique groups reads that stem from identical haplotypes. Within a clique, reads/contigs share (possibly low-frequency) true mutations while sequencing errors are not shared by the majority of reads (Figure 1, Panel B). Hence, cliques can be used to clearly distinguish between true mutations and sequencing errors. This further allows to correct these errors by transforming cliques into super-reads that represent an error-corrected consensus sequence of the reads in the clique.

Transitive edge removal. In graph-based assembly algorithms it is common to remove transitive edges. An edge $u \to w$ is called *transitive* if there exist a vertex v and edges $u \to v$, $v \to w$, illustrated in Figure 4, panel B. In de Bruijn graph-based approaches, genomes are reconstructed from paths in the graph, hence transitive edges should be removed before starting the assembly process. The first stage of our method, however, is essentially based on transitive edges: using such edges we find collections of overlapping reads, allowing us to correct errors in the sequencing reads.

But as coverage increases, the overlap graph G becomes more complicated and the number of maximal cliques grows rapidly. Therefore, we apply a procedure to remove *double transitive* edges. We call an edge $u \to w$ *double transitive* if there exists a vertex v and *transitive* edges $u \to v$, $v \to w$, illustrated in Figure 4, panel B. Note that, by definition, any double transitive edge is also single transitive. To find such edges, we first remove all non-transitive edges from the overlap graph to obtain the transitive graph G'. This can be done efficiently by computing the inner product of a_u^- and a_v^+ for all pairs $(u, v) \in V \times V$, where a_u^- (resp. a_v^+) is the adjacency vector of outgoing (resp. incoming) edges of u (resp. v). Applying this procedure to G we obtain G', and to find all double transitive edges we apply the same procedure to G'.

In each iteration of *Stage a*, we remove all double transitive edges from the overlap graph. This reduces the number of super-reads obtained in the first iteration by an order of magnitude, leading to a decrease in CPU time and memory usage of even two orders of magnitude (Supplemental Table S1). In *Stages b* and *c* the reads (contigs) are assumed to be already of high quality, so we remove not only double but also single transitive edges.

Read clustering. In *Stage a*, we cluster reads by enumerating maximal cliques in the overlap graph. After transitive edge removal the maximal clique size is 4 and the total number of cliques is polynomial in the number of nodes, so we can efficiently enumerate all maximal cliques. Here, we use of the degeneracy algorithm presented in (Eppstein et al. 2010). The first iteration of *Stage a* solely considers cliques of size of 4 to enable error correction, while subsequent iterations also consider maximal cliques of size 2 and 3.

In *Stages b* and *c*, after removing all (single and double) transitive edges, we merge pairs of contigs into new (extended) contigs. This does not require clique enumeration of any kind. See Figure 4, panel C for an illustration of the two read clustering techniques.

Super-read formation and error correction. As outlined above, we transform all reads/contigs within a clique (*Stages b, c*: pair of contigs) into a consensus sequence. The consensus base is determined by a position-wise weighted majority vote, where the weights correspond to the respective base quality scores, as described in (Töpfer et al. 2014) and Supplemental Methods. This procedure was designed to correct for all putative errors showing among members of a clique, which is especially relevant in the first iteration of *stage a*. In this specific iteration, therefore, we require cliques of size 4 and we remove the extremities of the resulting super-read (Figure 4, panel D). Reads that are not contained in any super-read are discarded after this iteration.

Graph updating. After constructing super-reads, we update the edges in the overlap graph on the super-reads. In *Stage a*, we do so by searching for common sub-reads among the super-reads. This approach is very efficient, but risks ignoring overlaps of super-reads that do not have an original read in common as a sub-read. In *Stages b* and *c* the graph is much sparser, so we then update the edges by considering all

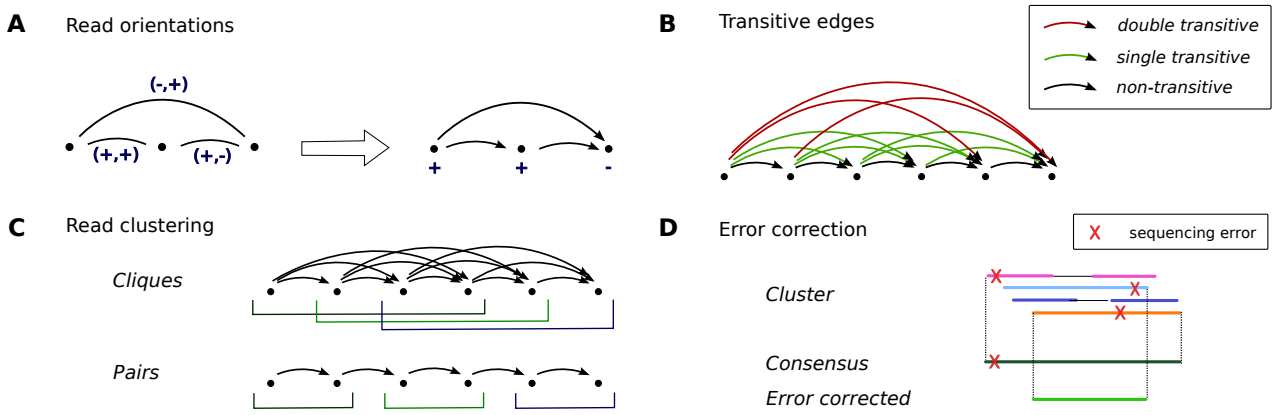

Figure 4: Algorithmic details. **A.** Read orientations: Given an edge $u \to v$ with orientations $(-,+)$. Then if u is labelled $+$, the induced label for v is $-$, while if u is labelled $-$ the induced label for v is $+$. This procedure leads to a vertex labeling in $O(V)$ time. **B.** Transitive edges: An edge $u \to w$ is called *single transitive* (resp. *double transitive*), shown in green (resp. red), if there exists a vertex v and edges (resp. *transitive* edges) $u \to v$, $v \to w$. **C.** Read clustering by cliques (top) or by pairs (bottom). **D.** Error correction: when a consensus sequence is constructed from a cluster of reads, the extremities are removed.

induced overlaps. This means that for every edge $u \to v$ in the original graph, we consider every overlap $u' \to v'$ for all $u' \in S_u, v' \in S_v$, where S_u, S_v consist of all super-reads containing u, v, respectively. In addition, we also reconsider all overlaps that did not make an edge in the original graph.

Iteration. The key idea of the SAVAGE assembly algorithm now is to repeatedly apply this twofold procedure of clique enumeration (*Stage a*) or merging pairs (*Stages b and c*) and super-read formation. Thereby, all super-reads of iteration $i \geq 1$ become nodes in the overlap graph of iteration $i+1$, which results in an overlap graph to be processed in iteration $i+1$. Key to succes of this idea is that super-reads (a.k.a. contigs) are constantly growing along the iterations, and, upon convergence, greatly exceed the length of the original reads. An example of the progression of contig lengths at the three stages of the algorithm is given in Supplemental Table S3.

Parameter settings

There are three parameters to be set, namely, the overlap score threshold δ, the mismatch rate mr allowed in the overlaps, and the minimal overlap length L. To analyze the behaviour of the overlap score function, we simulated 2x250 bp Illumina MiSeq reads from different genomes, diverging between 1% and 10%. We computed all overlaps among those reads and classified them by the number of true mutations in the overlap (not counting mismatches that are due to sequencing errors). This resulted in distributions P_i, $i \geq 0$, representing the overlap scores found in case of i true mutations (Supplemental Figure S2), from which we concluded that $\delta = 0.97$ is the optimal choice. To be more conservative, this threshold can be raised, but this comes at the cost of a decrease in the target genome coverage.

The mismatch rate parameter allows overlaps having a sufficiently high overlap score to become edges in the overlap graph if the mismatch rate is sufficiently low. By default, this parameter is set to 0, meaning that we only rely on the overlap score for constructing the overlap graph. However, when assembling master strains, the allowed mismatch rate was set to 0.01 such that strains diverging by less than 1% were merged into a consensus sequence.

Finally, the minimal overlap length was set to 60% of the average read length in *Stage a*. For *Stages*

b and *c*, we started at a rather high threshold of 500 bp, then reduced this to 150 bp, and eventually even lowered the threshold to 50 bp overlaps.

Frequency estimation

By keeping track of the original reads that are involved in a superread, we calculate the expected frequency of each haplotype constructed as the fraction of original reads contributing to this haplotype. Original reads that come from a conserved region of the viral quasispecies may contribute to multiple haplotypes; in this case the reads are weighted accordingly.

Given a contig C in the assembly $\mathcal{A}$, let $\text{len}(C)$ denote its length and let $\text{sub}(C)$ be the set of original reads that contributed to this contig. Furthermore, for each original read R, let α_R be the number of contigs that it is involved in. We define the weighted contributing read count of C as $w(C) = \sum_{R \in \text{sub}(C)} 1/\alpha_R$. Let $T(\mathcal{A})$ denote the total number of original reads contributing to the assembly, then we define the raw frequency as $\tilde{f}_C = w(C)/(T(\mathcal{A}) \cdot \text{len}(C))$. Here, the factor $\text{len}(C)$ in the denominator compensates for differences in contig lengths. Finally, we get the estimated frequency f_C of the contig by normalizing over all estimated frequencies: $f_C = \tilde{f}_C / \sum_{C \in \mathcal{A}} \tilde{f}_C$.

Other methods used for evaluation

For benchmarking, we compared SAVAGE against the state-of-the-art approaches ShoRAH (Zagordi et al. 2011) and PredictHaplo (Prabhakaran et al. 2014). Both methods were run with default parameter settings, after aligning the reads to the reference genome using BWA-MEM (Li 2013). The *de novo* assembler MLEHaplo (Malhotra et al. 2016b) required the reads to be error corrected first, for which we used MultiRes (Malhotra et al. 2016a) with default settings (recommended by the authors). Unfortunately, we could not compare against VGA (Mangul et al. 2014) and HaploClique (Töpfer et al. 2014) because these software packages were no longer maintained.

Data simulations

To evaluate performance of SAVAGE, we designed several simulated data sets. We used the software SimSeq (`https://github.com/jstjohn/SimSeq`) to simulate Illumina MiSeq reads from the genome of interest. In order to obtain reads similar to the real 5-virus-mix data, we simulated 2x250 bp paired-end reads, with a fragment size of 450 bp and the MiSeq error profile provided with the software.

Read trimming and merging

Before running any of the methods, the raw Illumina reads were trimmed using CutAdapt (Martin 2011). Next, we applied PEAR (Zhang et al. 2014a) for merging self-overlapping read pairs. This resulted in a final read set containing both single-end and paired-end reads, on which we ran SAVAGE. For the other methods (MLEHaplo, PredictHaplo, ShoRAH, and VICUNA) we used the trimmed reads without merging, since neither of these methods accepts a combination of single- and paired-end reads. In addition, MLEHaplo required an error correction step on the input reads which was performed using MultiRes (Malhotra et al. 2016a).

MetaQUAST evaluation

We use MetaQUAST (Mikheenko et al. 2016) for quality evaluation of the assembled contigs, which evaluates the contigs against each of the true viral genomes. By default, MetaQUAST uses the option

`--ambiguity-usage all`, which means that all possible alignments of a contig are taken into account. However, the genomes in a viral quasispecies can be so close together that a contig may align to multiple strains, even though it only matches one haplotype. Therefore, we manually changed this option to `--ambiguity-usage one`, such that for every contig only the best alignment is used. Contigs shorter than 500 bp were ignored during evaluation.

Data and software availability

All simulated data and a C++ implementation of SAVAGE are available for public use at `https://bitbucket.org/jbaaijens/savage`. The lab mix used for experiments can be downloaded from `https://github.com/cbg-ethz/5-virus-mix` and the ZIKV data are available in the NCBI Sequence Read Archive, experiment SRX1678783, run SRR3332513.

ACKNOWLEDGEMENTS

AS acknowledges funding through Vidi grant 639.072.309, provided by the Netherlands Organisation for Scientific Research (NWO). ER, AZ are supported by ANR Colib'read (ANR-12-BS02-0008), the Institut de Biologie Computationnelle (ANR-11-BINF-0002), and France Génomique.

DISCLOSURE DECLARATION

The authors declare that they have no conflict of interest.

References

Altschul S, Gish W, Miller W, Myers E, and Lipman D. 1990. Basic local alignment search tool. *J Mol Biol* **215**: 403–410.

Astrovskaya I, Tork B, Mangul S, Westbrooks K, Mandoiu I, Balfe P, and Zelikovsky A. 2011. Inferring viral quasispecies from 454 pyrosequencing reads. *BMC Bioinf* **12**: S1.

Bankevich A, Nurk S, Antipov D, Gurevich A, Dvorkin M, Kulikov A, Lesin V, Nikolenko S, Pham S, Prijbelski A, et al.. 2012. SPAdes: A new genome assembly algorithm and its applications to single-cell sequencing. *J Comp Biol* **19**: 455–477.

Bradnam K et al.. 2013. Assemblathon 2: evaluating de novo method of genome assembly in three vertebrate species. *GigaScience* **2**: 10.

Di Giallonardo F, Töpfer A, Rey M, Prabhakaran S, Duport Y, Leemann C, Schmutz S, Campbell N, Joos B, Lecca M, et al.. 2014. Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations. *Nucleic Acids Res* **42**: e115.

Domingo E, Sheldon J, and Perales C. 2012. Viral quasispecies evolution. *Microbiol Mol Biol Rev* **76**: 159–216.

Dudley DM, Aliota MT, Mohr EL, Weiler AM, Lehrer-Brey G, Weisgrau KL, Mohns MS, Breitbach ME, Rasheed MN, Newman CM, et al.. 2016. A rhesus macaque model of Asian-lineage Zika virus infection. *Nat Commun* **7**: 12204.

Duffy S, Shackelton LA, and Holmes EC. 2008. Rates of evolutionary change in viruses: patterns and determinants. *Nat Rev Genet* **9**: 267–276.

Eppstein D, Löffler M, and Strash D. 2010. Listing all maximal cliques in sparse graphs in near-optimal time. In *Proc. 21st Int. Symp. ISAAC*, volume 6506, pp. 403–414.

Ferragina P and Manzini G. 2000. Opportunistic data structures with applications. In *Proc. of FOCS*, pp. 390–398.

Gnerre S, MacCallum I, Przybylski D, Ribeiro F, Burton J, Walker B, Sharpe T, Hall G, Shea T, Sykes S, et al.. 2011. High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc Natl Acad Sci* **108**: 1513–1518.

Gregor I, Schönhuth A, and McHardy A. 2016. Snowball: Strain aware gene assembly of metagenomes. ArXiv:1510.03923.

Gurevich A, Saveliev V, Vyahhi N, and Tesler G. 2013. QUAST: quality assessment tool for genome assemblies. *Bioinformatics* **29**: 1072–1075.

Huang A, Kantor R, DeLong A, Schreier L, and Istrail S. 2012. QColors: An algorithm for conservative viral quasispecies reconstruction from short and non-contiguous next generation sequencing reads. *In Silico Biol* pp. 193–201.

Hunt M, Gall A, Ong SH, Brener J, Ferns B, Goulder P, Nastouli E, Keane JA, Kellam P, and Otto TD. 2015. IVA: accurate de novo assembly of RNA virus genomes. *Bioinformatics* **31**: 2374–2376.

Laserson J, Jojic V, and Koller D. 2011. Genovo: De novo assembly for metagenomes. *J Comp Biol* **18**: 429–443.

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with bwa-mem. ArXiv:1303.3997.

Luo R, Liu B, Xie Y, Li Z, Huang W, Yuan J, He G, Chen Y, Pan Q, Liu Y, et al.. 2012. SOAPdenovo2: an empirically improved memory-efficient short-read de novo assembler. *GigaScience* **1**: 18.

Malhotra R, Mukhopadhyay M, Poss M, and Acharya R. 2016a. A frame-based representation of genomic sequences for removing errors and rare variant detection in ngs data. ArXiv:1604.04803.

Malhotra R, Wu S, Mukhopadhyay M, Rodrigo A, Poss M, and Acharya R. 2016b. Maximum likelihood de novo reconstruction of viral populations using paired end sequencing data. ArXiv:1502.04239.

Mangul S, Wu N, Mancuso N, Zelikovsky A, Sun R, and Eskin E. 2014. Accurate viral population assembly from ultra-deep sequencing data. *Bioinformatics* **30**: i329–i337.

Martin M. 2011. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal* **17**: 10–12.

Mikheenko A, Saveliev V, and Gurevich A. 2016. Metaquast: evaluation of metagenome assemblies. *Bioinformatics* **32**: 1088–1090.

Nurk S, Meleshko D, Korobeynikov A, and Pevzner P. 2016. metaSPAdes: a new versatile de novo metagenomics assembler. Technical report, arXiv:1604.03071.

Prabhakaran S, Rey M, Zagordi O, Beerenwinkel N, and Roth V. 2014. HIV haplotype inference using a propagating dirichlet process mixture model. *IEEE Trans Comp Biol Bioinf* **11**: 182–191.

Prosperi MCF and Salemi M. 2012. Qure: software for viral quasispecies reconstruction from next-generation sequencing data. *Bioinformatics* **28**: 132–133.

Quince C, Lanzen A, Davenport RJ, and Turnbaugh PJ. 2011. Removing noise from pyrosequenced amplicons. *BMC Bioinf* **12**: 38.

Salzberg S, Phillippy A, Zimin A, Puiu D, Magoc T, Koren S, Treangen T, Schatz M, Delcher A, Roberts M, et al.. 2011. GAGE: A critical evaluation of genome assemblies and assembly algorithms. *Genome Res* **22**: 557–567.

Simpson J and Durbin R. 2012. Efficient de novo assembly of large genomes using compressed data structures. *Genome Res* **22**: 549–556.

Simpson J, Wong K, Jackman S, Schein J, Jones S, and Birol I. 2009. ABySS: a parallel assembler for short read sequence data. *Genome Res* **19**: 1117–1123.

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

Töpfer A, Zagordi O, Prabhakaran S, Roth V, Halperin E, and Beerenwinkel N. 2013. Probabilistic inference of viral quasispecies subject to recombination. *J Comp Biol* **20**: 113–123.

Välimäki N, Ladra S, and Mäkinen V. 2012. Approximate all-pairs suffix/prefix overlaps. *Inform Comput* **213**: 49–58.

Yang X, Charlebois P, Gnerre S, Coole M, Lennon N, Levin J, Qu J, Ryan E, Zody M, and Henn M. 2012. De novo assembly of highly diverse viral populations. *BMC Genomics* **13**: 475.

Zagordi O, Bhattacharya A, Eriksson N, and Beerenwinkel N. 2011. ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC Bioinf* **12**: 119.

Zagordi O, Geyrhofer L, Roth V, and Beerenwinkel N. 2010. Deep sequencing of a genetically heterogeneous sample: local haplotype reconstruction and read error correction. *J Comput Biol* **17**: 417–428.

Zerbino DR and Birney E. 2008. Velvet: algorithms for de novo short read assembly using de bruijn graphs. *Genome Res* **18**: 821–829.

Zhang J, Kobert K, Flouri T, and Stamatakis A. 2014a. Pear: A fast and accurate illumina paired-end read merge. *Bioinformatics* **30**: 614–620.

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