Long-read reconstruction of many diverse haplotypes with devider

Jim Shaw, Christina Boucher, Yun William Yu, et al.

Genome Res. 2025 35: 2637-2649 originally published online September 23, 2025
Access the most recent version at doi:10.1101/gr.280510.125

Method

Long-read reconstruction of many diverse haplotypes with devider

Jim Shaw,[1,2] Christina Boucher,[3] Yun William Yu,[4] Noelle Noyes,[5] and Heng Li[1,2]

[1]*Department of Data Science, Dana-Farber Cancer Institute, Boston, Massachusetts 02215, USA;* [2]*Department of Biomedical Informatics, Harvard Medical School, Boston, Massachusetts 02215, USA;* [3]*Department of Computer and Information Science and Engineering, University of Florida, Gainesville, Florida 32611, USA;* [4]*Ray and Stephanie Lane Computational Biology Department, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA;* [5]*Department of Veterinary Population Medicine, University of Minnesota, St. Paul, Minnesota 55421, USA*

Reconstructing exact haplotypes is important when sequencing a mixture of similar sequences. Long-read sequencing can connect distant alleles to disentangle similar haplotypes, but handling sequencing errors requires specialized techniques. Here, we present *devider*, an algorithm for haplotyping small sequences, such as viruses or genes, from long-read sequencing. devider uses a positional de Bruijn graph with sequence-to-graph alignment on an alphabet of informative alleles to provide a fast assembly-inspired approach compatible with various long-read sequencing technologies. On a synthetic Oxford Nanopore Technologies (ONT) long-read data set containing seven HIV strains, devider recovers 97% of the haplotype content and has the most accurate abundance estimates while taking <4 min and 1 GB of memory for >8000× coverage. Benchmarking on synthetic mixtures of antimicrobial-resistance (AMR) genes shows that devider recovers 83% of haplotypes, 23 percentage points higher than the next best method. On real Pacific Biosciences (PacBio) and ONT data sets, devider recapitulates previously known results in seconds, disentangling a bacterial community with more than 10 strains and an HIV-1 coinfection data set. We use devider to investigate the within-host diversity of a long-read bovine gut metagenome enriched for AMR genes, discovering 13 distinct haplotypes for a *tet(Q)* tetracycline-resistance gene with >18,000× coverage and six haplotypes for a *CfxA2* beta-lactamase gene. We find clear recombination blocks for these AMR gene haplotypes, showcasing devider's ability to unveil evolutionary signals for heterogeneous mixtures.

[Supplemental material is available for this article.]

The presence of highly similar genomic sequences within a single organism or a group of organisms is common in biological settings. Examples include viral quasispecies (Domingo and Perales 2019) in single-stranded RNA virus populations (e.g., HIV-1 and SARS-CoV-2) (Cuevas et al. 2015) or coexisting microbial subspecies in microbiomes (Van Rossum et al. 2020). Small genomic differences can have large functional implications (Vedantam et al. 1998; Olkkola et al. 2010), so it is crucial to disentangle this heterogeneity. We will call the recovery process of similar genomic sequences "haplotyping" or "phase." Although traditionally used in the context of diploid organisms, we extend the concept here to encompass the resolution of genetic diversity in microbes, viruses, or even genes.

With high-throughput sequencing, we can obtain haplotypes by linking reads that share informative alleles, for example, single-nucleotide polymorphisms (SNPs). Unfortunately, standard de novo short-read or long-read assembly approaches can collapse small-scale variation (Bickhart et al. 2022), returning only a consensus sequence. Although haplotype-resolved assembly has become standard for Pacific Biosciences (PacBio) HiFi sequencing (Cheng et al. 2021; Feng et al. 2022; Benoit et al. 2024; Li and Durbin 2024), HiFi data may not always be available, and assembly is computationally intensive. In contrast to assembly approaches, reference-based haplotyping uses a reference plus alignment to facilitate haplotyping; many existing approaches use the alignment, SNP calling, and then phasing paradigm (Lancia et al. 2001; Patterson et al. 2015; Edge et al. 2017; Feng et al. 2021; Knyazev et al. 2021; Cai and Sun 2022; Cai et al. 2022; Shaw and Yu 2022; Zhou et al. 2024).

We are interested in reference-based haplotyping for (1) long-read sequencing, (2) small sequences of approximately the read length, and (3) an *unknown*, possibly large, number of haplotypes. Reference-free approaches that tackle all or a subset of criteria 1–3 also exist (Baaijens et al. 2019; Luo et al. 2022), but the lack of reference adds additional algorithmic difficulties; we focus on the reference-based case. Long reads can connect more distant alleles across shared genomic regions compared with short reads. Still, a technical challenge is to deal with sequencing errors for certain technologies; for example, Oxford Nanopore Technologies (ONT) long reads can have 90%–99% sequencing accuracy depending on the chemistry and basecalling (Sereika et al. 2022). We focus on small sequences on the order of the read length (i.e., "local haplotyping") (Zagordi et al. 2011), but we do not necessarily require all reads to overlap the region of interest. This is sufficient for haplotyping genes of interest or estimating diversity. Although this seems like a simple task, systematic errors, high coverage, and low abundance haplotypes make accurate

Corresponding author: jshaw@ds.dfci.harvard.edu
Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.280510.125.

reconstruction challenging (Eliseev et al. 2020). Another class of approaches focuses on genome-scale haplotype reconstruction for prokaryotes (Vicedomini et al. 2021; Kazantseva et al. 2024; Shaw et al. 2024). However, these methods have not been tested for phasing high-diversity communities with more than five similar haplotypes.

Many haplotyping algorithms work for long reads or an unknown number of haplotypes, but only a few were designed to do both. RVHaplo (Cai and Sun 2022), HaploDMF (Cai et al. 2022), CliqueSNV (Knyazev et al. 2021), and iGDA (Feng et al. 2021) are long-read methods that call low-frequency SNPs and can phase diverse genomic sequences: RVHaplo uses a network clustering formulation; HaploDMF uses a matrix factorization approach; CliqueSNV uses a clique-merging approach; and iGDA uses a probabilistic local haplotyping step with an overlap-layout algorithm for reconstruction.

We present *devider*, a new long-read, reference-based haplotyping method for diverse small sequences. Given a set of aligned reads and SNPs, devider models the haplotyping problem as an assembly problem on a positional de Bruijn graph (PDBG). devider is inspired by the KSNP algorithm (Zhou et al. 2024), which uses a PDBG similarly but only for haplotyping diploids. We use the fact that the PDBG naturally splits if enough variation is present and collapses under ambiguity, thus haplotyping samples without prior knowledge of the number of distinct sequences. We then leverage the length of long reads by finding walks along the PDBG through unitig construction and read-to-graph alignment. We find that devider efficiently resolves haplotypes in a variety of synthetic and real data sets and show its versatility to reveal genomic heterogeneity.

Methods

At a high level, devider takes a set of aligned reads in BAM format plus SNPs in VCF format. It then outputs the sequences of SNPs for each recovered sequence (i.e., the haplotype), the abundance of each haplotype, and a base-level sequence for each haplotype. devider does not call SNPs or perform alignment, but we supply a wrapper for devider with minimap2 (Li 2018) and LoFreq (Wilm et al. 2012) for alignment and SNP calling. We use LoFreq for variant calling unless otherwise stated, but users can choose to use their own SNP caller.

devider follows in three major steps (Fig. 1A–C): (1) encoding the aligned reads with SNPs, (2) constructing a PDBG and positional unitig graph while filtering errors, and (3) aligning the SNP-encoded reads to the positional unitig graph to obtain walks along the graph that are supported by many reads, which represent candidate haplotypes.

Encoding reads with informative SNPs and filtering false positives

The first step is to filter false-positive SNPs arising from strand-specific errors. For each SNP, we apply a simple strand bias filter using Fisher's exact test (Guo et al. 2012) on the two-by-two contingency table defined by forward/reverse strands and reference/alternate alleles. For high coverage, Fisher's exact test is too stringent, so we also require an odds ratio of >1.5 or <1/1.5 for the contingency table. We then apply the Benjamini–Hochberg (Benjamini and Hochberg 1995) multiple-testing correction at 0.005 FDR threshold. More powerful methods are possible for strand filtering (McElroy et al. 2013), but we opt for a simple method because we are only concerned with phase.

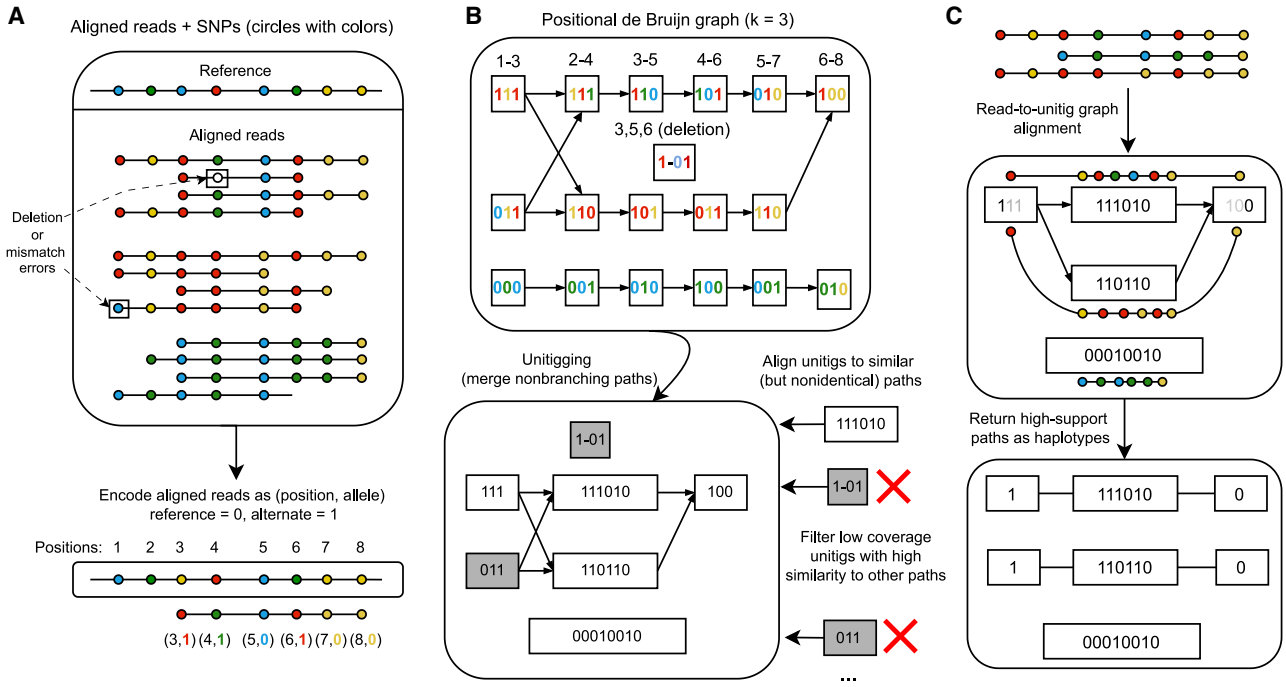

Figure 1. Algorithmic framework for devider. (*A*) Reads that are aligned to a reference are converted to a SNP representation with positional information. Sequencing errors lead to erroneous SNP encodings. (*B*) The SNP-encoded reads are turned into a positional de Bruijn graph (PDBG; $k = 3$ shown). In a PDBG, *k*-mers are collapsed if their alleles *and* their positions are identical. Errors in reads lead to spurious *k*-mers in the PDBG. After merging paths with in-degree and out-degree equal to one (unitigging), unitigs are aligned back to the graph to filter low-coverage, high-similarity unitigs. (*C*) Reads are aligned back to the filtered unitig graph to determine high-confidence walks through the graph. These paths are taken to be putative haplotypes. devider then postprocesses the haplotypes to output haplotype abundances, a base-level consensus of each haplotype, and the reads assigned to each haplotype.

We then subsample the SNPs if too many are present (i.e., the sequences are highly divergent) as follows: For a given α (for discussion, see section "Practical details and implementation") and median number of SNPs contained in a read β, we downsample uniformly to retain only α/β of the SNPs if $\alpha < \beta$. We do this because when too many SNPs are present, a group of SNPs may span only a small region of the genome, which we wish to avoid in our subsequent SNP-based *k*-mer approach. We then realign each read using blockaligner (Liu and Steinegger 2023) against the 32 bp flanks around each filtered SNP site, replacing the site with all possible alleles and then selecting the allele that gives the highest alignment score.

Finally, we encode each read as an ordered list of tuples such as (3,1), (4,1), (5,0), (6,1), (7,0), (8,0) (see Fig. 1A). The first number represents the *i*th SNP in the reference to which the read is aligned, and the second number indicates the allele that the read contains, where zero is the reference and one to three indicate the alternates. We skip SNPs if either the SNP site has a deletion in the read or the read's base at the SNP site is neither an alternate nor reference allele in the VCF. Thus, a sequence such as (3,1), (5,0) is possible.

PDBG and uniting on the SNP alphabet

All reads will now be considered SNP-encoded reads. An SNP-encoded read is a string in an alphabet $\Sigma = Z^+ \times \{0, 1, 2, 3\}$ subject to the constraint that the first symbol in each letter is in increasing order; for example, (5,1) must come before (7,0). *k*-mers of SNP-encoded reads are defined as elements in Σ^k, and we construct a de Bruijn graph in the usual way by collecting all *k*-mers within the reads and adding directed edges between *k*-mers that overlap $k-1$ SNPs (Fig. 1B, top). However, because the positions are encoded in the alphabet Σ, we only collapse *k*-mers if they have the same positions. This is now a *positional* de Bruijn graph (PDBG) (Ronen et al. 2012; Bao et al. 2014; Cameron et al. 2017). The PDBG is a directed acyclic graph (DAG) because any cycle would violate the increasing ordering of the SNPs. This fact will be important for the subsequent sequence-to-graph alignment steps.

We automatically choose a value of *k* as follows. Let γ be the 33rd percentile of the number of SNPs contained in a read, and let N be the number of SNPs in the reference (after filtering). Let M be a parameter representing the maximum possible value of *k*, which is set to avoid long error-prone *k*-mers. This depends on the sequencing technology and is picked through a preset option. We let $k = \min(M, N \cdot \frac{3}{4}, \gamma)$. We do not let *k* span more than 75% of the reference to not miss *k*-mers if a smaller haplotype only covers a subsection of the reference. We discuss the parameter choices of M in the section "Practical details and implementations" and show results over varying values of *k* in the Results.

We apply an initial filtering step to discard the likely erroneous *k*-mers. Let the coverage of a *k*-mer be the number of times it appears in a SNP-encoded read, and let m be the mean *k*-mer coverage. Let A be the minimum allowable abundance for a haplotype (default = 0.0025 or 0.25%). We filter *k*-mers that appear only once or have less than $m \cdot A$ coverage. Finally, from the filtered PDBG, we construct the positional unitig graph by merging all nonbranching paths into unitigs (Fig. 1B, bottom) (Myers 2005). We let the coverage of the unitig be the mean coverage of the merged *k*-mers.

Filtering unitigs by error-aware unitig-to-graph alignment

The main technical challenge is to simplify the unitig graph, which can still have many spurious unitigs arising from *k*-mer errors (Supplemental Figs. 1, 2). In the standard de Bruijn graph assembly, tip removal and bubble popping are used to remove noise and variation (Li et al. 2015). However, we do not want to remove the true variation. Thus, we use a unitig-to-graph alignment approach plus coverage information for fine-grained unitig filtering; this generalizes both tip removal and bubble popping but uses alignment information and coverage information.

Classifying errors

Let G be the positional unitig graph. Recall that a unitig node v can be represented by $v = (x_1, x_2, \ldots, x_n)$, where $x_i = (a_i, b_i)$ with a_i the SNP position and b_i the allele. Given two unitigs v_1 and v_2, we classify errors as SNP deletions (del), reference-to-alternate (rtoa) mismatches, and alternate-to-reference (ator) mismatches as follows:

- $s(v_1, v_2)$ is the number of SNPs that are the same in v_1 and v_2 (i.e., share the same position and base).
- $del(v_1, v_2)$ is the number of SNPs in v_1 that are deleted relative to v_2 (i.e., do not appear in v_2 and lie between the first and last SNP of v_2).
- $rtoa(v_1, v_2)$ and $ator(v_1, v_2)$ represent the number of SNPs that have the same position but different alleles between v_1 and v_2. Specifically, $rtoa(v_1, v_2)$ is the number of different SNPs for which v_1 has the *reference* allele (and thus v_2 has an alternate allele), whereas *ator* is the number of differing SNPs for which v_1 has an *alternate* allele.

We stratify the error types because they appear with different frequencies. *rtoa* and *ator* differ owing to reference bias: If a read comes from a haplotype with a true alternate allele, it may systematically align with the reference allele incorrectly (Stevenson et al. 2013). A SNP deletion (*del*) error is the most common owing to the following reason: Consider a biallelic site with two alleles (A, C) and a read originating from the haplotype with allele A. devider's convention is to consider the read's SNP to be deleted if there is a base-level deletion in the CIGAR string or the read's base is G or T at the SNP site. Of the four error possibilities (deletion, substitution to C, substitution to G, substitution to T), three of them result in a SNP deletion in the SNP-encoded reads. Furthermore, long reads can also have inherently higher deletion error rates (Delahaye and Nicolas 2021).

Alignment to DAG

Next, we align the unitigs back to the unitig graph to find an alignment path. However, we disallow alignments back to unitig itself, because this would always be the best match. If there is a path such that *s* (successes) is large relative to *del*, *rtoa*, and *ator* (errors) and this path has much higher coverage than the unitig, then the unitig is probably an error that originates from the path.

By abuse of notation, let $s(v,P)$ refer to the number of matching alleles between v and the string spelled out by a unitig path P. We wish to find a path that does not contain v and maximizes $s(v,P)$. Furthermore, we impose that (1) ties are broken by taking the highest coverage path, (2) v's first and last SNP positions must be within P's first and last SNP positions, and (3) all unitigs in P overlap v. We do not need to penalize for the error terms in this step because more matching alleles imply fewer deletions and errors.

Because of the DAG structure of the PDBG (and thus the positional *unitig* graph), the optimal path can be found with standard dynamic programming. Taking a unitig v and a topological order on nodes $1,\ldots, n$ that overlap v, we wish to find the optimal path ending at node v_j. Let P_j be a path that ends at node v_j. The

following recurrence holds:

$$\text{score}(j) := \max_{P_j} s(v, P_j)$$
$$= \max_{v_\ell \in \text{in}(v_j)} \left[\max_{P_\ell} s(v, P_\ell) + s(v, v_j) - s(v, \text{overlap}(v_j, v_l)) \right].$$

Note that we subtract the overlap to avoid double counting, because v_j and its incoming unitigs may overlap. After obtaining all scores, we find optimal paths that satisfy the three constraints above. Each alignment takes $O(|V||E|)$ worst-case running time, but in practice, unitig graphs are sparse (e.g., Supplemental Fig. 1), so this step is not a bottleneck.

Error-aware filtering

Once we have a best path P for each unitig v, we use a one-sided binomial test based on coverage to filter spurious unitigs. This filtering is "error-aware" in the sense that different types of errors in the sequence-to-graph alignment have different frequencies (see section "Classifying errors"), so we wish to differentiate the types of errors within the binomial test. We run the alignment and error-aware filter starting from the smallest coverage unitig, remove unitigs from the graph, and repeat for the next smallest coverage unitig.

Let $cov(v)$ be the unitig coverage, and let $cov(P)$ be (1) the highest coverage unitig in P that completely covers v or (2) the mean unitig coverage in P if no unitigs in P completely cover v. We filter out v if $\Pr(cov(v) \geq Binomial(cov(P),q)) < 0.005$, where q is defined as follows: define $p_{del} = 0.35$, $p_{rtoa} = 0.15$, and $p_{ator} = 0.10$ to model the error frequency as previously discussed. Then $q = p_{ator}^{ator(v,P)} \cdot p_{rtoa}^{rtoa(v,P)} \cdot \delta(p_{del})$, where $\delta(p_{del}) = p_{del}$ if $ator(v, P) + rtoa(v, P) = 0$ but otherwise is equal to one. This formula slightly deviates from the independence assumption of the three error modalities. We found that systematic biases could occasionally cause two errors to occur nonindependently. Thus, we loosen the independence of $p(del)$, which was the main cause of errors. In general, we found that these values are conservative and work for error-prone reads (~95%) but could likely be tightened to improve sensitivity in the future as sequencing technologies improve.

Read-to-graph alignment and collecting haplotype paths

To find candidate haplotype paths through the filtered unitigs, we align reads back to the graph using a sequence-to-DAG alignment algorithm (Fig. 1C), which is similar to the unitig-to-graph alignment. We will use these alignments to find well-supported walks through the PDBG, representing candidate haplotypes.

Let r be a read. Here, rather than finding a path to maximize $s(r,P)$ as we did before, we find a path to maximize $s(r,P) - 3 \times err(r, P)$, where $err = ator(r, P) + rtoa(r, P)$. The penalty term of three was chosen to penalize errors more than similarities. Although true deleted SNPs in a haplotype are possible, we assume deletions occur mainly owing to noise, so we do not penalize deletions. Compared to unitig-to-graph alignment, we add two changes. First, we allow P to bridge sinks-to-sources. This allows the alignment to rescue broken paths owing to erroneous filtering. Second, we do not require the path to completely cover the read, also in case of erroneous filtering. We use the same dynamic programming procedure as for unitig-to-graph alignment; $err(r, P)$ can also be split into the exact same recurrence and is thus solvable by dynamic programming.

After aligning all reads, we have a set of paths with read-alignment multiplicities. If two or more equally good paths exist for a read, we do not assign the read to any path. If a path is contained within another, we remove the contained path. If the path is *uniquely* contained in another, we add the contained path's multiplicity to the noncontained path's multiplicity. We filter for high-confidence paths by removing paths for which the read-alignment multiplicity is <3× the minimum unitig coverage within the path.

Haplotype consensus and outputs

For the filtered paths, we assign the reads to each path by finding the path that maximizes $2 * s(r, P) - 3 * rtoa(r, P) - 5 * ator(r, P) - del(r, P)$, assigning the read to no path if the score is less than zero. The penalty weights are heuristically chosen to reflect the frequency of occurrence: Deletions are common so they are penalized less, and $rtoa(r, P)$ is penalized less than $ator(r, P)$ because of reference bias; and the read r will preferentially carry the reference allele owing to biases in aligning to references. We then take the consensus allele for each SNP site after assignment and the abundance as the fraction of assigned reads.

Lastly, we perform a deduplication step as follows. For some resolution parameter ρ, we merge two resulting consensus haplotypes H_1, H_2 together if $(rtoa^{\Delta}(H_1, H_2) + ator^{\Delta}(H_1, H_2))/s^{\Delta}(H_1, H_2) < \rho$, where Δ considers only unambiguous SNPs for which the >0.75 fraction of reads carries the majority allele. After merging, we reassign the reads to the best candidate haplotype subject to the same scoring scheme above, take consensuses, and filter low-abundance and low-depth haplotypes (default = 0.25% and 5× depth). We iterate this procedure until the number of haplotypes does not change. We then return the abundances, the reads assigned to each haplotype, and the sequences of the SNPs for each haplotype. Finally, we also output a majority *base-level* consensus sequence of all bases, not just SNPs, for each set of assigned reads by iterating through alignments in the BAM file. We return the "N" base if the fraction of reads supporting the majority base is less than a parameter (default = 0.66).

Practical details and implementation

devider is implemented in Rust and uses rust-htslib (Bonfield et al. 2021) and rust-bio crates (Köster 2016). We implemented the following preset options to help users pick parameters: `old-long-reads`, `nanopore-r9`, `nanopore-r10`, `hi-fi`. These correspond to $(M, \alpha, \rho) = (10, 50, 0.02)$, $(20, 150, 0.01)$, $(35, 250, 0.005)$, and $(100, 500, 0.001)$, respectively. The current default is `nanopore-r9`. We wrap devider in minimap2 and LoFreq in the `run_devider_pipeline` script in the repository. This runs minimap2, SAMtools, and LoFreq to generate an indexed BAM and VCF pair. For LoFreq, we found that disabling base-alignment qualities with the `-B` option improved sensitivity on nanopore reads. All other parameters were set to default.

Results

We show our benchmarking setups in Supplemental Figures 3 and 4. For a set of genomes, we use badread (Wick 2019) to simulate nanopore long reads with default settings except for lengths and accuracy, which we make explicit for each data set in the following. We picked an arbitrary reference genome and ran all tools with this reference genome. We then compared their predicted haplotypes with the true haplotypes. We benchmarked devider against iGDA v1.0.1, CliqueSNV v2.0.3, RVHaplo v2, and HaploDMF (version May 2022). We tried running Strainline, a de novo viral quasispecies assembly method, but it produced an error that was related to an unresolved GitHub issue (https://github.com/HaploKit/Strainline/issues/17). We ran all methods with default settings and iGDA in its ONT setting with the `ont_context_effect_read_qv_12_base_qv_12` model. We ran all the methods with 10 threads. We used the following metrics for benchmarking:

1. Hamming SNP error is the mean percentage of incorrect SNPs for each predicted haplotype against its best-matching genome.
2. Fraction recovered is the fraction of SNPs recovered for all genomes after matching predicted haplotypes against their best-matching genomes.
3. Earth mover's distance (EMD) (Rubner et al. 1998) is a measure of the distance between the predicted abundances and the true abundances. The pairwise distance function we used for the EMD is the number of mismatched SNPs between the predicted and true haplotype, and the weights are the predicted relative abundances.
4. Haplotype error is the predicted number of haplotypes minus the true number.

HIV-1 benchmarking (seven strains plus varying coverage and two to 30 strains)

HIV-1 serves as a standard benchmarking genome for viral quasispecies methods owing to its fast mutation rate as an ssRNA virus and its clinical and public health importance. Thus, we created two HIV-1 communities: a seven-strain staggered abundance community and multiple communities from two to 30 strains with uniform abundances. These strains ranged from 99.27% to 99.71% pairwise nucleotide similarity. Here, we define "strain" to be a distinct reference genome that we are trying to reconstruct.

Seven strains at staggered abundances

We took a set of 30 HIV-1 genomes from Kinloch et al. (2023) with accessions available in Supplemental Table 1. For the first seven-strain data set, we selected an arbitrary reference (OR483991.1) and the seven most similar strains as determined by skani (Shaw and Yu 2023). The abundances were staggered at a 1:3:5:7:9:10:20 ratio, with the smallest strain coverage ranging from 3× to 160×. We simulated reads in three settings with (accuracy, length) = (95%, 9000 bp), (98%, 9000 bp), and (95%, 3000 bp) with a length standard deviation of 500 bp. The precision of 95% represents older or faster nanopore sequencing runs, whereas 98% is more representative of the best current basecalling/chemistries (Sereika et al. 2022). HIV-1 genomes are ∼9000 bp, so the two length settings represent complete and partial coverage.

On the 95% accuracy, 9000 bp data set (Fig. 2A), devider and HaploDMF performed the best. devider had slightly worse mean fraction recovered (97.7% for devider vs. 98.8% for HaploDMF), equal mean haplotype error (−0.15 for devider vs. +0.15 for HaploDMF), and better EMD (0.41 for devider vs. 1.66 for HaploDMF). However, devider was more than 13 times faster and took less than 10 times less memory than HaploDMF on average. Only devider and iGDA achieved perfect SNP Hamming errors across all data points. The greatest performance difference was at low coverage; at 3× minimum coverage, devider estimated six correct haplotypes and HaploDMF estimated eight (incorrectly outputting an additional haplotype), but the other methods only recovered one, one, and three haplotypes for RVHaplo, CliqueSNV, and iGDA respectively. We found that CliqueSNV consistently obtained six haplotypes, missing the low abundance haplotype. We tried to increase its sensitivity by lowering its abundance threshold to 0.25% (the same as devider), but it drastically overestimated the number of haplotypes, outputting more than 30 haplotypes at high coverage. To show that devider is robust to parameter choices on this data set, we varied the *k*-mer length between 10 to 30 and found that devider still had the best haplotype error, fraction recovered, and EMD (Supplemental Fig. 5).

On the 95%, 3000 bp data set (Supplemental Fig. 6A), devider had the second-best mean fraction recovered (92.3% vs. 93.7% for HaploDMF) and the best mean EMD (1.18 vs. 4.18 for the second-best HaploDMF). devider had the smallest absolute mean haplotype error (0.46 vs. 0.53 for the second-best HaploDMF) compared with the other methods. However, the mean Hamming SNP error for devider (0.37%) was slightly worse than that for iGDA (0.06%) and RVHaplo (0.07%) but better than that for HaploDMF (1.02%).

On the 98% accuracy, 9000 bp data set (Supplemental Fig. 6B), devider and HaploDMF had near-perfect performances for mean fraction recovered (98.9% vs. 99.9% respectively) and haplotype error (−0.08 vs. −0.15 respectively). devider had perfect Hamming SNP error (0% vs. 0.024% for iGDA, the second best) and the best EMD (0.31 vs. 2.3 for HaploDMF, the second best). de Bruijn graph methods work well with lower error rates because the probability of having an error within a *k*-mer is approximately $(\text{accuracy})^k$; this is encouraging for future nanopore data sets as read accuracy increases.

Two to 30 strains at uniform coverages

We simulated reads at 95% accuracy and 9000 bp average length for two to 30 strains from the same set of HIV-1 genomes, with each genome at 15×–145× coverage uniformly at random (Fig. 2B). devider, iGDA, and HaploDMF performed better than RVHaplo and CliqueSNV on this data set. devider was the best method on all metrics except HaploDMF recovered slightly more haplotypes on average (10.2 vs. 10.1 for devider). Although no method could capture all strains when more than 10 were present, devider consistently had a low EMD, implying that missed strains were either lowly abundant or highly similar. Unlike HaploDMF, RVHaplo, and CliqueSNV, devider had a lower EMD as the number of strains increased. In fact, devider had a smaller EMD with 30 strains than with two strains. CliqueSNV also performed well when the number of strains was fewer than 10 (81.7% mean fraction recovered vs. 83.6% for devider), but its performance dropped when more than 10 strains were present (37.5% mean fraction recovered for 10–20 strains vs. 65.6% for devider).

devider and iGDA stood out in terms of efficiency compared with the other methods. iGDA and devider took 5 and 23 sec on average, respectively, and <0.5 GB of RAM. Note that we included lofreq's runtime and memory usage in devider's results. RVHaplo took 370 sec, HaploDMF took 790, and CliqueSNV 820 sec on average. CliqueSNV was efficient except for the case with 30 strains, when the runtime ballooned to 10,956 sec, which corresponds to ∼3.04 h. We found that this was because CliqueSNV sets a 3 h (10,800 sec) time limit by default if it cannot solve the haplotyping problem, after which it outputs no haplotypes.

SARS-CoV-2 minor haplotyping

We next investigated the ability of algorithms to detect minor haplotypes at uneven abundances. To do this, we created multiple two-strain synthetic mixtures of Delta (accession MZ009823.1) and Omicron (accession OL672836.1) SARS-CoV-2 genomes, with the minor Omicron strain ranging from 0.39% to 25% abundance. This setup represents a situation in which we would like to detect a circulating, low-abundance strain with high sequence similarity (99.58% between these two genomes) prior to emergence. This was exactly the case with SARS-CoV-2 evolution after the

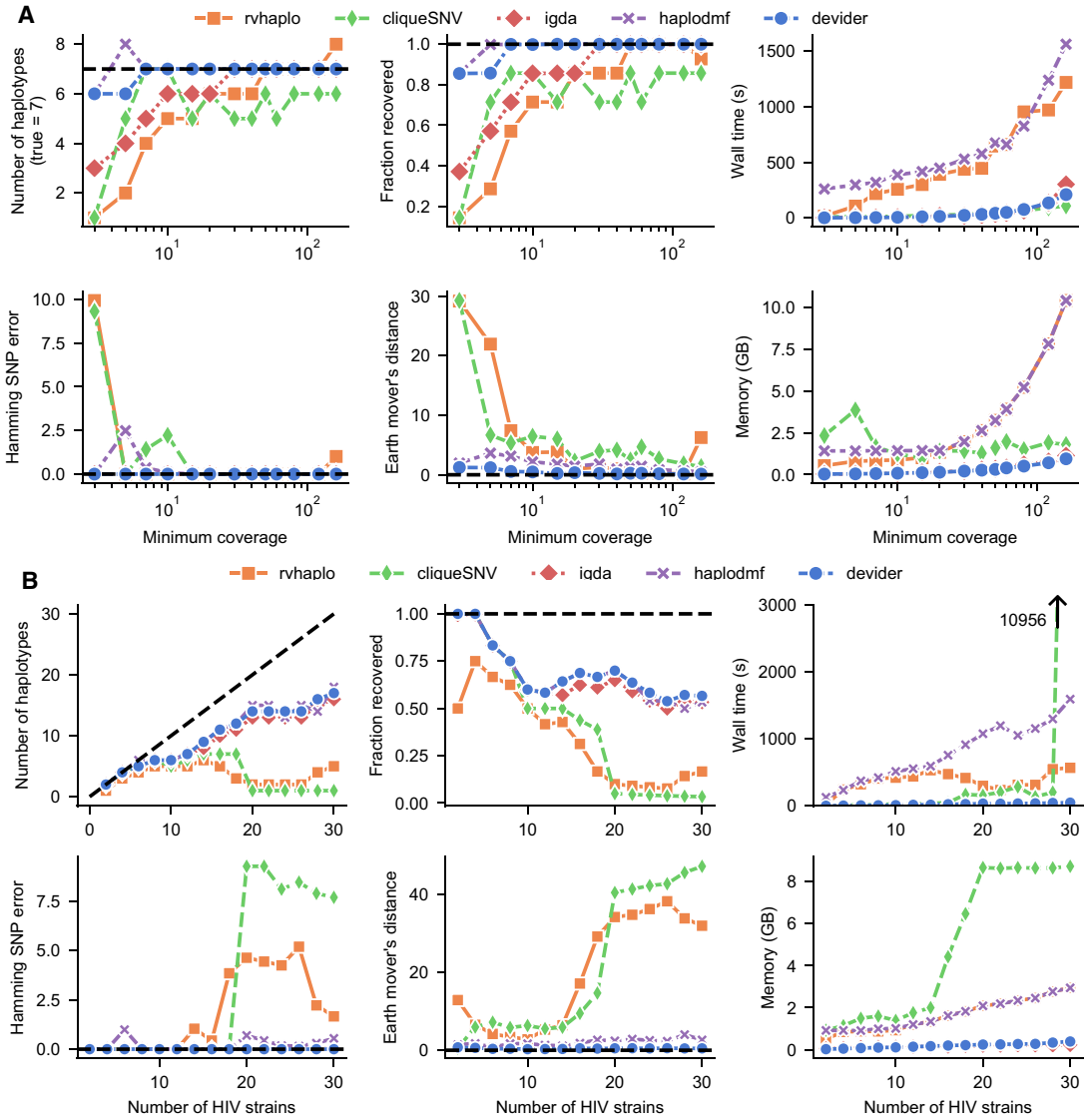

Figure 2. Benchmarking long-read haplotyping tools on simulated HIV-1 communities. (*A*) Seven HIV-1 strains from Kinloch et al. (2023) at staggered abundances (1:3:5:7:9:10:20) with simulated reads (9000 bp mean length; 95% mean accuracy). The *x*-axis indicates the depth of coverage for the lowest-coverage strain. (*B*) Two to 30 HIV-1 strains with 15×–145× uniformly random coverage and the same simulated read length/accuracy. CliqueSNV failed to complete within the default 3 h time limit for the last sample. iGDA does not output abundances, so its earth mover's distance could not be computed. Dashed black lines indicate optimal performance.

Delta phase of the pandemic: a pattern of selective sweeps and emergence of a single low-abundance strain characterized viral strain frequencies (Boyle et al. 2022; Markov et al. 2023).

We first simulated near full-length reads (30 kb mean length with 500 bp standard deviation) and 95% accuracy at 3000× total coverage; that is, a 1% abundant haplotype would have 30× coverage (Fig. 3A). devider and RVHaplo were the best two methods in this data set, successfully constructing exactly two haplotypes in 3/8 and 5/8 of the mixtures, respectively. The limit of detection of RVHaplo was 1.56%, slightly better than devider at 3.12%, and devider incorrectly output a spurious low abundance haplotype at 25% abundance. iGDA did not estimate two haplotypes in any abundances for this data set; HaploDMF output incorrect numbers of haplotypes except at 25% abundance; and CliqueSNV failed to detect the minor haplotype below 12.5% abundance.

In practice, capturing the entire SARS-CoV-2 in a single read is difficult, and approaches focus on amplicon sequencing of shorter target regions (Tyson et al. 2020). For a more realistic test, we focus on the spike protein region of SARS-CoV-2, a smaller region of ~4 kbp for which there are long-read amplicon protocols (Liao et al. 2022; Nimsamer et al. 2023). We simulated reads for the spike protein region in the same genomes at 3000× total coverage but now with 4 kb mean length and 500 bp standard deviation (Fig. 3B). In this setup, devider and RVHaplo were again the best two methods, but this time they had almost identical performance at a detection limit of 3.12%. HaploDMF was able to recover the minor haplotype at 1.56% abundance, but HaploDMF also outputs an

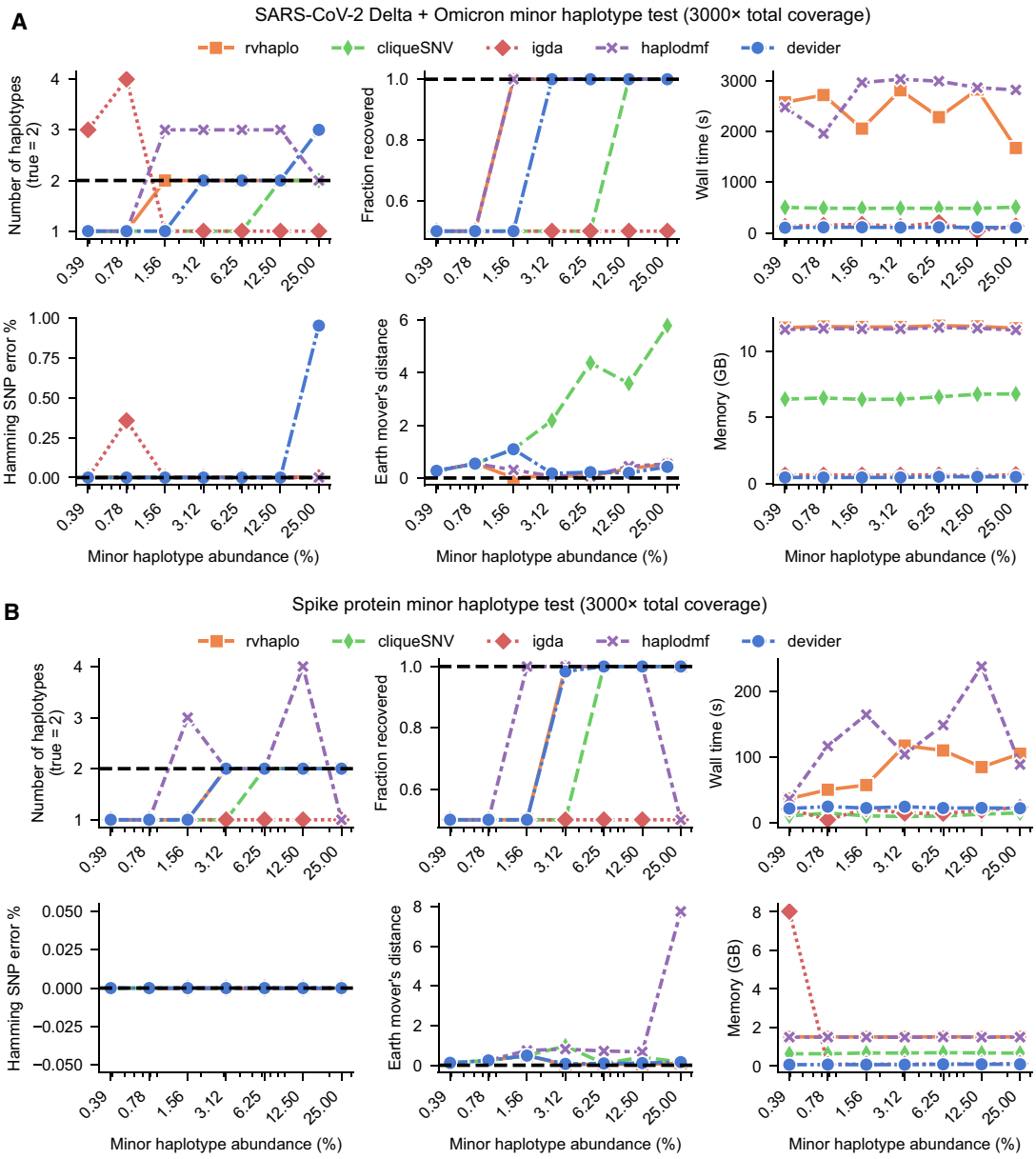

Figure 3. Benchmarking for two-strain SARS-CoV-2 synthetic sequencing mixtures at varying abundances. (*A*) Results for mixtures of full-length reads of Delta (major) and Omicron (minor) genomes. (*B*) Results for mixtures of reads that only cover the spike protein gene for the Delta and Omicron genomes.

additional spurious haplotype. In general, detecting haplotypes of very low abundance in abundances <2% with 95% accuracy reads is a difficult task, and RVHaplo was the best method with devider in second.

Synthetic AMR gene haplotyping for 53 AMR gene groups

We chose AMR genes as a gene-level haplotyping benchmark owing to their diversity in sequence composition, length, biological significance, and also the prevalence of targeted enrichment sequencing protocols (Slizovskiy et al. 2022; Baba et al. 2023; Shay et al. 2023) for which our methods are potentially usable. We clustered AMR genes in the MEGARes database version 3.0 (Bonin et al. 2023) by computing all pairwise average nucleotide identities (ANIs) and mean alignment fractions (AFs) with skani (using the --slow preset) and then using the Leiden algorithm (Traag et al. 2019; Camargo et al. 2024) with edge weights as $ANI^* AF$ at resolution 1.00. Of the remaining 53 clusters with 15 or more different haplotypes, we sampled between two and 15 haplotypes at coverage 80×–1000×, both uniformly at random. We then simulated reads at 95% accuracy and 1500 bp length with 200 bp standard deviation. The lengths of the AMR genes ranged from 721 to 3303 bp.

In this data set, devider was the best method for the mean haplotype error, SNP Hamming error, and fraction recovered (Fig. 4A), with −1.5, 0.06%, and 83. 3%, respectively. CliqueSNV was the next best method with −4.2, 0.34%, and 60.2% on the same metrics. To investigate the discrepancy between different methods, we stratified the fraction recovered by coverage, percentage identity of haplotype to reference, and abundance (Fig. 4B). As

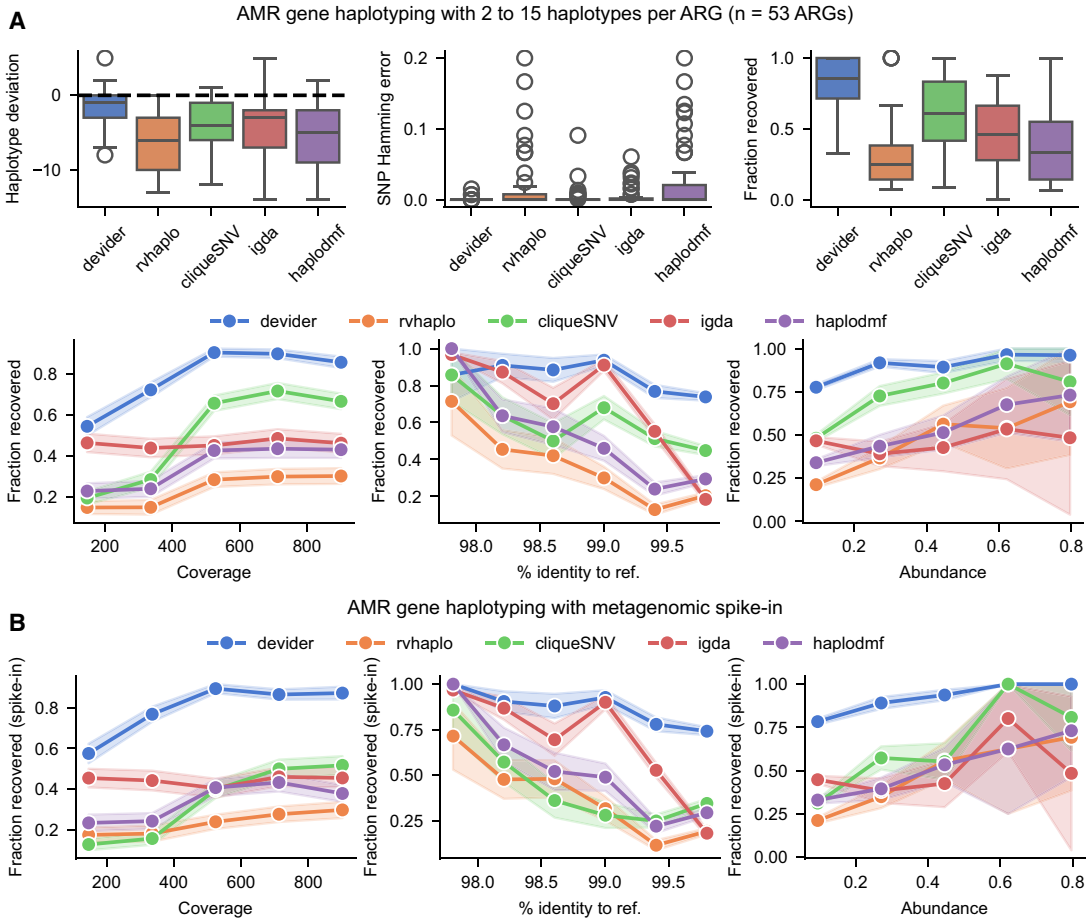

Figure 4. Benchmarking for haplotyping of synthetic mixtures of antimicrobial resistance (AMR) genes. (*A*, *top*) Results for 53 sets of AMR genes with two to 15 haplotypes and reads simulated at 80×–1000× coverage (95% identity; 1500 mean length), both picked uniformly at random. (*Bottom*) The same results but with fraction recovered as a function of the haplotype's coverage, its % nucleotide identity to the reference, and its abundance (i.e., normalized coverage). (*B*) Fraction recovered for the AMR genes after spiking the AMR reads into a synthetic long-read mouse gut metagenome from CAMI2. Each method was rerun after aligning the pooled data set against the AMR gene references. Error bars indicate standard errors after binning data points along the *x*-axis. Box plots show the median, the 25th and 75th percentiles, and 1.5× the interquartile range.

expected, lower coverage, which is correlated with lower abundance, leads to a lower fraction recovered for all the methods. We found that the likely cause of the performance discrepancy was high similarity haplotypes: more than 2/3 of the haplotypes had >99.5% similarity to the reference, and in these haplotypes, the recovered fraction of devider was 73.8% compared with CliqueSNV with 46.3%, the next best method. Thus, devider is the most sensitive method for highly similar haplotypes, which is important because capturing even small variations in AMR genes can alter phenotypes (Vedantam et al. 1998).

AMR data set with spike-in metagenome

We extended the previous AMR haplotyping experiment to a metagenomics setting. In metagenomics, the input is a mixture of microbial genomic reads *and* reads from AMR genes. Thus, we mixed the previously simulated AMR reads into a simulated long-read mouse gut metagenome from CAMI2 (labeled as sample 0) (Meyer et al. 2022), hereafter referred to as the spiked metagenome.

In particular, genomes within the CAMI2 metagenome contain AMR genes or sequences with homology with AMR genes. Mapping the CAMI2 reads against MEGARES with minimap2 resulted in 130 AMR genes with >2× coverage but only 10 AMR genes with >20× coverage. Thus, the spike-in metagenome contains the previously simulated AMR haplotypes (with two to 15 haplotypes and 80×–1000× coverage) as well as this new tail of low-abundance AMR haplotypes from the CAMI2 metagenome.

In this setup (shown in Supplemental Fig. 4), we measured the ability of each method to recover the original simulated AMR haplotypes (Fig. 4B). devider, iGDA, HaploDMF, and RVHaplo had a <2% difference in fraction recovered compared with the previous (without spike-in) case. However, CliqueSNV fell to 42.9% (with spike-in) from 60.2% (no spike-in). Thus, devider can recover abundant haplotypes for a long-tailed abundance distribution with low abundance haplotypes, a common characteristic of metagenomic data.

Results on real heterogeneous sequencing mixtures

HIV-1 coinfection haplotyping

Mori et al. (2022) sequenced a set of HIV-1 samples with full-length nanopore amplicons (5.8%–7% sequencing error rate)

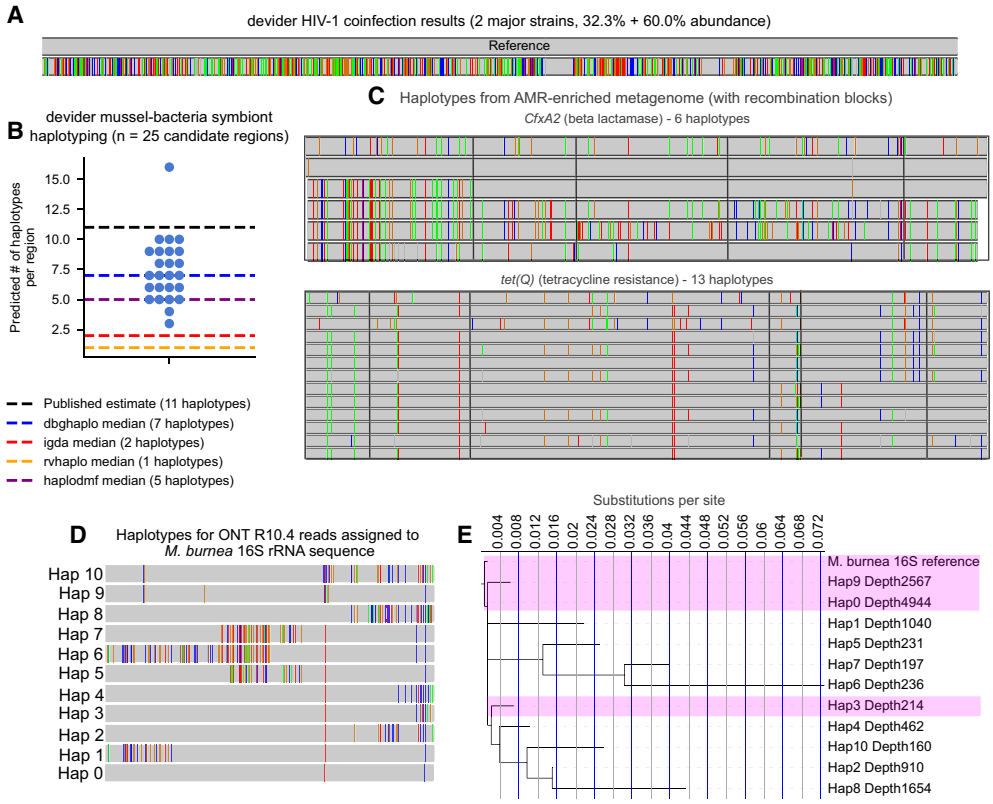

Figure 5. Long-read haplotyping results from real samples subjected to a variety of sequencing technologies. (*A*) Haplotypes from long-read HIV-1 nanopore sequencing (93%–94.2% predicted sequencing accuracy) of an HIV coinfection from Mori et al. (2022). Two major haplotypes were found by devider, confirming previous results. Mismatched bases are shown with the reference as the upper haplotype. (*B*) Haplotyping results for PacBio RS II sequencing (89.5% mean gap-compressed identity against reference) of an intracellular bacterial symbiont community within deep-sea mussels from Ansorge et al. (2019), who predicted 11 strains to be present. Twenty-five candidate single-copy regions with high SNP diversity were haplotyped by devider, iGDA, and RVHaplo; CliqueSNV was excluded because it timed out on multiple regions. devider produced higher diversity estimates compared with iGDA and RVHaplo, which both produced three or fewer haplotypes across all sites. (*C*) devider haplotyping of a long-read bovine gut metagenome enriched for AMR genes. *CfxA2* (3200× coverage) and *tet(Q)* (19,500× coverage and last 1000 bp shown) haplotype sequences with >30× coverage and 1% abundance are shown with mismatches against their reference sequences in MEGARes v3.0. Mismatches shared by all haplotypes are removed. Recombination blocks are outlined in black as predicted by GARD. (*D*) devider haplotypes from an ONT R10.4 16S rRNA data set for the reference 16S sequence of the most abundant species *Massilia burnea*. (*E*) Phylogenetic tree of haplotypes assigned to the *M. burnea* reference. Depth of coverage is shown next to the haplotype ID. The *x*-axis shows the branch length from the root. Highlighted haplotypes have >99% identity to the reference.

and detected a possible HIV-1 coinfection for a patient (labeled TRN9) by stochastically subsampling reads and reassembling. Here, we ran devider, iGDA, RVHaplo, and CliqueSNV on these reads and an arbitrary reference (NC_001802.1) to try to confirm their results. devider gave four haplotypes (60.0%, 32.34%, 4.12%, and 3.57% abundance) and RVHaplo gave two (67% and 32.9% abundance). CliqueSNV only output one haplotype, and iGDA outputs no haplotypes.

Mori et al. (2022) found two majority haplotypes, which is corroborated by RVHaplo and devider. We show devider's two majority haplotypes in Figure 5A. We investigated the two additional minority haplotypes output by devider to see if they were erroneous; curiously, these two haplotypes were a mix of the two majority haplotypes and were well supported by "chimeric" reads with noisy breakpoints (Supplemental Fig. 7). HIV-1 is known to recombine, which may be a possible explanation, but in vitro PCR recombination could also be a possibility (Meyerhans et al. 1990). Porechop (Wick et al. 2017) found only 6/2500 reads with an adapter in the middle, so an erroneous ligation is not likely. We do not speculate further on the biology, but we note that these chimeric reads represent a real signal that devider can retrieve for further biological investigation.

Mussel-bacteria symbiont community with 11 estimated strains

Ansorge et al. (2019) investigated the strain-level diversity of an intracellular, sulfur-oxidizing bacterial symbiont community within deep-sea mussels. They sequenced one sample using the PacBio RS II, which produces high-error long reads. Ansorge et al. (2019) do not explicitly define "strain" but calculate it via two methods: (1) counting unique structural arrangements of single-copy marker genes from long-red assemblies and (2) using a viral quasispecies detection algorithm (Jayasundara et al. 2015) for short-read sequencing in the same data set. We investigated whether long-read haplotyping without assembly could give similar estimates, even with noisy reads (mean gap-compressed identity against reference = 89.5%). To generate reasonable diversity estimates, we haplotyped 3 kb regions of the reference genome (GCF_900128535) that had more than 10 SNPs (as detected by LoFreq) and coverage between 200 and 250. These criteria were chosen because the

average read length was 3.8 kbp and the estimated single-copy coverage, calculated by dividing the total read bases by the genome size, was 220. We ran all the methods on these regions in PacBio mode (if such a preset existed) and the `old-long-reads` preset for devider.

In these 25 regions, devider estimated a median of seven haplotypes (Fig. 5B). RVHaplo estimated a median of five haplotypes, but iGDA and RVHaplo only estimated two or fewer median haplotypes. CliqueSNV failed to run on multiple regions, timing out after 3 h and outputting nothing, so we did not include its results. devider estimated 16 haplotypes for one region, and subsequent investigation revealed many additional alignments near the edge of contigs. The first 600 bases had mean coverage >300×, possibly indicating a duplicated region (Supplemental Fig. 8) for a subset of the strains. Ultimately, devider was able to capture some of the known diversity in these samples using noisy long reads, whereas most other methods failed.

Discovering recombinant AMR genes in AMR-enriched long-read metagenomes

We used devider to haplotype an AMR-enriched PacBio CCS long-read fecal metagenome from a cow that received intensive antibiotic treatments (Slizovskiy et al. 2022). We first dereplicated the MEGARes v3.0 database at 95% using vsearch (Rognes et al. 2016) to avoid ambiguous mapping of reads to highly similar genes. We then ran devider with extra stringent parameters, setting the minimum abundance to 1% and minimum depth to 30×.

In total, we found 18 different dereplicated AMR genes with two or more haplotypes. Of these 18 genes, nine were tetracycline-resistance genes that were phased into 52 distinct haplotypes. The highest coverage gene was *tet(Q)* (Lacroix and Walker 1996) at 19,500× coverage and was phased into 13 haplotypes. Other high-diversity genes we found included mefA (Daly et al. 2004), a macrolide efflux pump, at 7100× coverage, that was phased into 12 haplotypes, as well as *CfxA2* (Iwahara et al. 2006), a beta-lactamase, at 3200× coverage, that was phased into six haplotypes. We illustrate the haplotypes of *tet(Q)* and *CfxA2* in Figure 5C (Integrative Genomics Viewer [IGV]) (Robinson et al. 2011) screenshots in Supplemental Figs. 9, 10). We found a distinct mosaic structure within these haplotypes, suggesting a history of recombination within these haplotypes. We used MAFFT (Katoh et al. 2002) to generate a multiple sequence alignment from devider's haplotypes and GARD (Kosakovsky Pond et al. 2006) to detect recombination, which found evidence of recombination for both genes. We draw breakpoints at which GARD's model-averaged support was >0.3 in Figure 5C. The consensus haplotypes were well supported by the reads: the six haplotypes for *CfxA2* had 68%, 72%, 90%, 71%, 44%, and 70% of their assigned reads spanning all four recombination breakpoints (in order from top to bottom of Fig. 5C). Across all alleles, a median of 99% of the reads within the haplotype supported the consensus allele, indicating confident haplotypes. Mosaicism owing to recombination is a well-documented characteristic of some ribosomal protection proteins including *tet(Q)* (Warburton et al. 2016), and *CfxA* genes are commonly colocalized with an element known to play a role in the mosaic behavior of conjugative elements (García et al. 2008). Thus, these detected recombination events are supported by known mechanisms in these two AMR genes.

Disentangling 16S rRNA amplicon sequences from R10.4 nanopore data

As an additional use case, we investigated using devider as a reference-based method to cluster full-length 16S rRNA amplicon sequences from ONT R10.4 sequencing, the newest and most accurate chemistry. Currently, computational profilers for ONT 16S sequencing align amplicons directly to reference genomes (Curry et al. 2022). Denoising algorithms for generating amplicon sequencing variants (ASVs) requires a significant fraction of error-free reads (Callahan et al. 2016), so they are still not usable for the newest R10.4 reads. We investigated whether devider clusters can be used to generate reference-based ASVs, allowing for species-level identification. We profiled a 16S R10.4 soil sample from Zhang et al. (2023) (obtained from the NCBI Sequence Read Archive [SRA; https://www.ncbi.nlm.nih.gov/sra] under accession number SRR23176498) as follows. We first used Emu (Curry et al. 2022) to quantify species-level abundances and then applied devider with Emu's default 16S database. We used the same pipeline for devider except parameters `-B 2 -A 3 -s 20` for minimap2 (more aggressive extension) and `--mapq-cutoff 1 --supp-mapq-cutoff 1 --min-qual 20` for devider (MAPQ is low for 16S databases; higher base quality thresholds for newer nanopore reads). devider only took 200 sec for the entire data set, excluding alignment and variant calling.

For the most abundant species, *Massilia eburnea* (18.6% abundance from emu), we visualized the 11 haplotypes found by devider and a phylogenetic tree (Letunic and Bork 2021) of the consensus sequences constructed by FastTree (Price et al. 2010) and MAFFT (Fig. 5D,E). Each consensus had >90× depth of coverage and were well supported by reads (Supplemental Fig. 11). Only Hap0, Hap3, and Hap9 had >99% identity to the reference, a suggested threshold for species-level assignment (Edgar 2018). These three haplotypes had 59.9% combined abundance, so using this 99% identity threshold, 40.1% of the sequences should be considered novel species-level 16S sequences. As a more extreme example, *Vicinamibacter silvestris* was the third most abundant species according to Emu (4.5% abundance); however, every one of the devider consensuses had <95% identity to the reference (Supplemental Fig. 12). Ultimately, robustly generating ASVs is a highly nontrivial task that we do not claim to solve. However, our investigation shows how devider can be a useful tool for curating reference-based ASVs from deep, heterogeneous long-read amplicon sequences.

Discussion

We presented devider, a method for retrieving high-similarity haplotypes from long-read sequencing of heterogeneous sequences. devider leverages a PDBG assembly approach on a subset of informative alleles to disentangle variation. This framework is efficient and naturally resolves variation without the need to explicitly infer the number of haplotypes. The key technical challenge was to remove sequencing artifacts within the PDBG, especially for error-prone long reads while retaining high sensitivity, which we accomplished through an error-aware sequence-to-graph alignment approach.

Based on our benchmarks and experience, we found devider to excel for heterogeneous and high depth samples. Key examples include amplicon sequencing, enriched metagenome sequencing, or viral sequencing. Anecdotally, we have found devider to also work for high-abundance species in unenriched long-read metagenomes. In general, devider can work for <10× depth (Fig. 2A), but its sensitivity increases for higher depth. Currently, we do not try to recover haplotypes with <0.25% abundance and <5× depth by default. In practice, the exact detection limits will be some function of relative abundance, depth, and sequence divergence (see

Fig. 4). We have shown that devider can distinguish up to approximately 20 distinct haplotypes in benchmarks and real data, although more could be possible depending on the relative divergence of haplotypes.

We designed devider to work with a wider range of technologies and sequencing error rates. We limited devider to reconstructing "small" sequences on the order of read length for conservative recovery. However, we showed that synthetic reconstructing an HIV genome of even more than three times the mean read length is possible as long as some of the reads are long enough. As error rates improve, it may be possible to attempt a longer haplotype reconstruction using our approach.

A key limitation is our reference-based approach, which is unable to recover new sequences de novo. However, reference-based approaches are intrinsically more efficient and simpler than de novo approaches. We believe that reference-based methods are complementary to de novo approaches. As the capabilities and the need for resolved sequences at the haplotype level from long reads continue to increase, devider will be a fast and useful tool for retrieving accurate haplotypes.

Software availability

devider is open source and is available on GitHub (https://github.com/bluenote-1577/devider) or bioconda (Grüning et al. 2018) and as Supplemental Code. The scripts for reproducing our figures are available at GitHub (https://github.com/bluenote-1577/devider-test) and as Supplemental Scripts.

Competing interest statement

The authors declare no competing interests.

Acknowledgments

N.N. is supported by the National Institutes of Health (NIH), National Institute of Allergy and Infectious Diseases (NIAID) grant 1R01AI173928-01A1. C.B. is supported by the NIH NIAID grant 5R01AI141810-02. Y.W.Y. is supported by the US National Science Foundation, Division of Biological Infrastructure grant 2531433. H.L. is supported by the National Institutes of Health, National Human Genome Research Institute grant R01HG 010040. J.S. is supported by a Natural Sciences and Engineering Research Council of Canada Postdoctoral Fellowship (PDF-587396-2024).

Author contributions: C.B., Y.W.Y., N.N., and J.S. conceived the study. J.S. designed and implemented the method with supervision from H.L. J.S. performed the bioinformatics analysis and benchmarking with supervision from all authors. All authors contributed to the writing of the manuscript.

References

Ansorge R, Romano S, Sayavedra L, Porras MÁG, Kupczok A, Tegetmeyer HE, Dubilier N, Petersen J. 2019. Functional diversity enables multiple symbiont strains to coexist in deep-sea mussels. *Nat Microbiol* **4:** 2487–2497. doi:10.1038/s41564-019-0572-9

Baaijens JA, Van der Roest B, Köster J, Stougie L, Schönhuth A. 2019. Full-length de novo viral quasispecies assembly through variation graph construction. *Bioinformatics* **35:** 5086–5094. doi:10.1093/bioinformatics/btz443

Baba H, Kuroda M, Sekizuka T, Kanamori H. 2023. Highly sensitive detection of antimicrobial resistance genes in hospital wastewater using the multiplex hybrid capture target enrichment. *mSphere* **8:** e0010023. doi:10.1128/msphere.00100-23

Bao E, Jiang T, Girke T. 2014. AlignGraph: algorithm for secondary de novo genome assembly guided by closely related references. *Bioinformatics* **30:** i319–i328. doi:10.1093/bioinformatics/btu291

Benjamini Y, Hochberg Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Ser B Methodol* **57:** 289–300. doi:10.1111/j.2517-6161.1995.tb02031.x

Benoit G, Raguideau S, James R, Phillippy AM, Chikhi R, Quince C. 2024. High-quality metagenome assembly from long accurate reads with metaMDBG. *Nat Biotechnol* **42:** 1378–1383. doi:10.1038/s41587-023-01983-6

Bickhart DM, Kolmogorov M, Tseng E, Portik DM, Korobeynikov A, Tolstoganov I, Uritskiy G, Liachko I, Sullivan ST, Shin SB, et al. 2022. Generating lineage-resolved, complete metagenome-assembled genomes from complex microbial communities. *Nat Biotechnol* **40:** 711–719. doi:10.1038/s41587-021-01130-z

Bonfield JK, Marshall J, Danecek P, Li H, Ohan V, Whitwham A, Keane T, Davies RM. 2021. HTSlib: C library for Reading/writing high-throughput sequencing data. *GigaScience* **10:** giab007. doi:10.1093/gigascience/giab007

Bonin N, Doster E, Worley H, Pinnell LJ, Bravo JE, Ferm P, Marini S, Prosperi M, Noyes N, Morley PS, et al. 2023. MEGARes and AMR++, v3.0: an updated comprehensive database of antimicrobial resistance determinants and an improved software pipeline for classification using high-throughput sequencing. *Nucleic Acids Res* **51:** D744–D752. doi:10.1093/nar/gkac1047

Boyle L, Hletko S, Huang J, Lee J, Pallod G, Tung HR, Durrett R. 2022. Selective sweeps in SARS-CoV-2 variant competition. *Proc Natl Acad Sci* **119:** e2213879119. doi:10.1073/pnas.2213879119

Cai D, Sun Y. 2022. Reconstructing viral haplotypes using long reads. *Bioinformatics* **38:** 2127–2134. doi:10.1093/bioinformatics/btac089

Cai D, Shang J, Sun Y. 2022. HaploDMF: viral haplotype reconstruction from long reads via deep matrix factorization. *Bioinformatics* **38:** 5360–5367. doi:10.1093/bioinformatics/btac708

Callahan BJ, McMurdie PJ, Rosen MJ, Han AW, Johnson AJA, Holmes SP. 2016. DADA2: high-resolution sample inference from Illumina amplicon data. *Nat Methods* **13:** 581–583. doi:10.1038/nmeth.3869

Camargo AP, Call L, Roux S, Nayfach S, Huntemann M, Palaniappan K, Ratner A, Chu K, Mukherjeep S, Reddy TBK, et al. 2024. IMG/PR: a database of plasmids from genomes and metagenomes with rich annotations and metadata. *Nucleic Acids Res* **52:** D164–D173. doi:10.1093/nar/gkad964

Cameron DL, Schröder J, Penington JS, Do H, Molania R, Dobrovic A, Speed TP, Papenfuss AT. 2017. GRIDSS: sensitive and specific genomic rearrangement detection using positional de Bruijn graph assembly. *Genome Res* **27:** 2050–2060. doi:10.1101/gr.222109.117

Cheng H, Concepcion GT, Feng X, Zhang H, Li H. 2021. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nat Methods* **18:** 170–175. doi:10.1038/s41592-020-01056-5

Cuevas JM, Geller R, Garijo R, López-Aldeguer J, Sanjuán R. 2015. Extremely high mutation rate of HIV-1 in vivo. *PLoS Biol* **13:** e1002251. doi:10.1371/journal.pbio.1002251

Curry KD, Wang Q, Nute MG, Tyshaieva A, Reeves E, Soriano S, Wu Q, Graeber E, Finzer P, Mendling W, et al. 2022. Emu: species-level microbial community profiling of full-length 16S rRNA Oxford Nanopore sequencing data. *Nat Methods* **19:** 845–853. doi:10.1038/s41592-022-01520-4

Daly MM, Doktor S, Flamm R, Shortridge D. 2004. Characterization and prevalence of MefA, MefE, and the associated *msr*(D) gene in *Streptococcus pneumoniae* clinical isolates. *J Clin Microbiol* **42:** 3570–3574. doi:10.1128/JCM.42.8.3570-3574.2004

Delahaye C, Nicolas J. 2021. Sequencing DNA with nanopores: troubles and biases. *PLoS One* **16:** e0257521. doi:10.1371/journal.pone.0257521

Domingo E, Perales C. 2019. Viral quasispecies. *PLoS Genet* **15:** e1008271. doi:10.1371/journal.pgen.1008271

Edgar RC. 2018. Updating the 97% identity threshold for 16S ribosomal RNA OTUs. *Bioinformatics* **34:** 2371–2375. doi:10.1093/bioinformatics/bty113

Edge P, Bafna V, Bansal V. 2017. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. *Genome Res* **27:** 801–812. doi:10.1101/gr.213462.116

Eliseev A, Gibson KM, Avdeyev P, Novik D, Bendall ML, Pérez-Losada M, Alexeev N, Crandall KA. 2020. Evaluation of haplotype callers for next-generation sequencing of viruses. *Infect Genet Evol* **82:** 104277. doi:10.1016/j.meegid.2020.104277

Feng Z, Clemente JC, Wong B, Schadt EE. 2021. Detecting and phasing minor single-nucleotide variants from long-read sequencing data. *Nat Commun* **12:** 3032. doi:10.1038/s41467-021-23289-4

Feng X, Cheng H, Portik D, Li H. 2022. Metagenome assembly of high-fidelity long reads with hifiasm-meta. *Nat Methods* **19:** 671–674. doi:10.1038/s41592-022-01478-3

García N, Gutiérrez G, Lorenzo M, García JE, Píriz S, Quesada A. 2008. Genetic determinants for cfxA expression in bacteroides strains isolated from human infections. *J Antimicrob Chemother* **62:** 942–947. doi:10.1093/jac/dkn347

Grüning B, Dale R, Sjödin A, Chapman BA, Rowe J, Tomkins-Tinch CH, Valieris R, Köster J. 2018. Bioconda: sustainable and comprehensive software distribution for the life sciences. *Nat Methods* **15:** 475–476. doi:10.1038/s41592-018-0046-7

Guo Y, Li J, Li CI, Long J, Samuels DC, Shyr Y. 2012. The effect of strand bias in Illumina short-read sequencing data. *BMC Genomics* **13:** 666. doi:10.1186/1471-2164-13-666

Iwahara K, Kuriyama T, Shimura S, Williams DW, Yanagisawa M, Nakagawa K, Karasawa T. 2006. Detection of *cfxA* and *cfxA2*, the *β*-lactamase genes of *Prevotella* spp., in clinical samples from dentoalveolar infection by real-time PCR. *J Clin Microbiol* **44:** 172–176. doi:10.1128/JCM.44.1.172-176.2006

Jayasundara D, Saeed I, Maheswararajah S, Chang B, Tang SL, Halgamuge SK. 2015. ViQuaS: an improved reconstruction pipeline for viral quasispecies spectra generated by next-generation sequencing. *Bioinformatics* **31:** 886–896. doi:10.1093/bioinformatics/btu754

Katoh K, Misawa K, Kuma K, Miyata T. 2002. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Res* **30:** 3059–3066. doi:10.1093/nar/gkf436

Kazantseva E, Donmez A, Frolova M, Pop M, Kolmogorov M. 2024. Strainy: phasing and assembly of strain haplotypes from long-read metagenome sequencing. *Nat Methods* **21:** 2034–2043. doi:10.1038/s41592-024-02424-1

Kinloch NN, Shahid A, Dong W, Kirkby D, Jones BR, Beelen CJ, MacMillan D, Lee GQ, Mota TM, Sudderuddin H, et al. 2023. HIV reservoirs are dominated by genetically younger and clonally enriched proviruses. *mBio* **14:** e02417–23. doi:10.1128/mbio.02417-23

Knyazev S, Tsyvina V, Shankar A, Melnyk A, Artyomenko A, Malygina T, Porozov YB, Campbell EM, Switzer WM, Skums P, et al. 2021. Accurate assembly of minority viral haplotypes from next-generation sequencing through efficient noise reduction. *Nucleic Acids Res* **49:** e102. doi:10.1093/nar/gkab576

Kosakovsky Pond SL, Posada D, Gravenor MB, Woelk CH, Frost SD. 2006. GARD: a genetic algorithm for recombination detection. *Bioinformatics* **22:** 3096–3098. doi:10.1093/bioinformatics/btl474

Köster J. 2016. Rust-Bio: a fast and safe bioinformatics library. *Bioinformatics* **32:** 444–446. doi:10.1093/bioinformatics/btv573

Lacroix JM, Walker CB. 1996. Detection and prevalence of the tetracycline resistance determinant Tet Q in the microbiota associated with adult periodontitis. *Oral Microbiol Immunol* **11:** 282–288. doi:10.1111/j.1399-302X.1996.tb00182.x

Lancia G, Bafna V, Istrail S, Lippert R, Schwartz R. 2001. SNPs problems, complexity, and algorithms. In *Algorithms—ESA 2001. Lecture Notes in Computer Science* (ed. auf der Heide FM), pp. 182–193. Springer, Berlin.

Letunic I, Bork P. 2021. Interactive Tree of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. *Nucleic Acids Res* **49:** W293–W296. doi:10.1093/nar/gkab301

Li H. 2018. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34:** 3094–3100. doi:10.1093/bioinformatics/bty191

Li H, Durbin R. 2024. Genome assembly in the telomere-to-telomere era. *Nat Rev Genet* **25:** 658–670. doi:10.1038/s41576-024-00718-w

Li D, Liu CM, Luo R, Sadakane K, Lam TW. 2015. MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph. *Bioinformatics* **31:** 1674–1676. doi:10.1093/bioinformatics/btv033

Liao YC, Chen FJ, Chuang MC, Wu HC, Ji WC, Yu GY, Huang TS. 2022. High-integrity sequencing of spike gene for SARS-CoV-2 variant determination. *Int J Mol Sci* **23:** 3257. doi:10.3390/ijms23063257

Liu D, Steinegger M. 2023. Block Aligner: an adaptive SIMD-accelerated aligner for sequences and position-specific scoring matrices. *Bioinformatics* **39:** btad487. doi:10.1093/bioinformatics/btad487

Luo X, Kang X, Schönhuth A. 2022. Strainline: full-length de novo viral haplotype reconstruction from noisy long reads. *Genome Biol* **23:** 29. doi:10.1186/s13059-021-02587-6

Markov PV, Ghafari M, Beer M, Lythgoe K, Simmonds P, Stilianakis NI, Katzourakis A. 2023. The evolution of SARS-CoV-2. *Nat Rev Microbiol* **21:** 361–379. doi:10.1038/s41579-023-00878-2

McElroy K, Zagordi O, Bull R, Luciani F, Beerenwinkel N. 2013. Accurate single nucleotide variant detection in viral populations by combining probabilistic clustering with a statistical test of strand bias. *BMC Genomics* **14:** 501. doi:10.1186/1471-2164-14-501

Meyer F, Fritz A, Deng ZL, Koslicki D, Lesker TR, Gurevich A, Robertson G, Alser M, Antipov D, Beghini F, et al. 2022. Critical assessment of metagenome interpretation: the second round of challenges. *Nat Methods* **19:** 429–440. doi:10.1038/s41592-022-01431-4

Meyerhans A, Vartanian JP, Wain-Hobson S. 1990. DNA recombination during PCR. *Nucleic Acids Res* **18:** 1687–1691. doi:10.1093/nar/18.7.1687

Mori M, Ode H, Kubota M, Nakata Y, Kasahara T, Shigemi U, Okazaki R, Matsuda M, Matsuoka K, Sugimoto A, et al. 2022. Nanopore sequencing for characterization of HIV-1 recombinant forms. *Microbiol Spectr* **10:** e01507–22. doi:10.1128/spectrum.01507-22

Myers EW. 2005. The fragment assembly string graph. *Bioinformatics* **21 Suppl 2:** ii79–ii85. doi:10.1093/bioinformatics/bti1114

Nimsamer P, Sawaswong V, Klomkliew P, Kaewsapsak P, Puenpa J, Poovorawan Y, Payungporn S. 2023. "Nano COVID-19": nanopore sequencing of spike gene to identify SARS-CoV-2 variants of concern. *Exp Biol Med* **248:** 1841–1849. doi:10.1177/15353702231190931

Olkkola S, Juntunen P, Heiska H, Hyytiäinen H, Hänninen ML. 2010. Mutations in the *rpsL* gene are involved in streptomycin resistance in *Campylobacter coli*. *Microbial Drug Resistance* **16:** 105–110. doi:10.1089/mdr.2009.0128

Patterson M, Marschall T, Pisanti N, van Iersel L, Stougie L, Klau GW, Schönhuth A. 2015. WhatsHap: weighted haplotype assembly for future-generation sequencing reads. *J Comput Biol* **22:** 498–509. doi:10.1089/cmb.2014.0157

Price MN, Dehal PS, Arkin AP. 2010. FastTree 2: approximately maximum-likelihood trees for large alignments. *PLoS One* **5:** e9490. doi:10.1371/journal.pone.0009490

Robinson JT, Thorvaldsdóttir H, Winckler W, Guttman M, Lander ES, Getz G, Mesirov JP. 2011. Integrative genomics viewer. *Nat Biotechnol* **29:** 24–26. doi:10.1038/nbt.1754

Rognes T, Flouri T, Nichols B, Quince C, Mahé F. 2016. VSEARCH: a versatile open source tool for metagenomics. *PeerJ* **4:** e2584. doi:10.7717/peerj.2584

Ronen R, Boucher C, Chitsaz H, Pevzner P. 2012. SEQuel: improving the accuracy of genome assemblies. *Bioinformatics* **28:** i188–i196. doi:10.1093/bioinformatics/bts219

Rubner Y, Tomasi C, Guibas L. 1998. A metric for distributions with applications to image databases. In *Sixth International Conference on Computer Vision (IEEE cat. no. 98CH36271)*, Bombay, India, pp. 59–66. IEEE, Piscataway, NJ.

Sereika M, Kirkegaard RH, Karst SM, Michaelsen TY, Sørensen EA, Wollenberg RD, Albertsen M. 2022. Oxford Nanopore R10.4 long-read sequencing enables the generation of near-finished bacterial genomes from pure cultures and metagenomes without short-read or reference polishing. *Nat Methods* **19:** 823–826. doi:10.1038/s41592-022-01539-7

Shaw J, Yu YW. 2022. flopp: extremely fast long-read polyploid haplotype phasing by uniform tree partitioning. *J Comput Biol* **29:** 195–211. doi:10.1089/cmb.2021.0436

Shaw J, Yu YW. 2023. Fast and robust metagenomic sequence comparison through sparse chaining with skani. *Nat Methods* **20:** 1661–1665. doi:10.1038/s41592-023-02018-3

Shaw J, Gounot JS, Chen H, Nagarajan N, Yu YW. 2024. Floria: fast and accurate strain haplotyping in metagenomes. *Bioinformatics* **40:** i30–i38. doi:10.1093/bioinformatics/btae252

Shay JA, Haniford LSE, Cooper A, Carrillo CD, Blais BW, Lau CHF. 2023. Exploiting a targeted resistome sequencing approach in assessing antimicrobial resistance in retail foods. *Environ Microbiome* **18:** 25. doi:10.1186/s40793-023-00482-0

Slizovskiy IB, Oliva M, Settle JK, Zyskina LV, Prosperi M, Boucher C, Noyes NR. 2022. Target-enriched long-read sequencing (TELseq) contextualizes antimicrobial resistance genes in metagenomes. *Microbiome* **10:** 185. doi:10.1186/s40168-022-01368-y

Stevenson KR, Coolon JD, Wittkopp PJ. 2013. Sources of bias in measures of allele-specific expression derived from RNA-seq data aligned to a single reference genome. *BMC Genomics* **14:** 536. doi:10.1186/1471-2164-14-536

Traag VA, Waltman L, van Eck NJ. 2019. From Louvain to Leiden: guaranteeing well-connected communities. *Sci Rep* **9:** 5233. doi:10.1038/s41598-019-41695-z

Tyson JR, James P, Stoddart D, Sparks N, Wickenhagen A, Hall G, Choi JH, Lapointe H, Kamelian K, Smith AD, et al. 2020. Improvements to the ARTIC multiplex PCR method for SARS-CoV-2 genome sequencing using nanopore. bioRxiv doi:10.1101/2020.09.04.283077

Van Rossum T, Ferretti P, Maistrenko OM, Bork P. 2020. Diversity within species: interpreting strains in microbiomes. *Nat Rev Microbiol* **18:** 491–506. doi:10.1038/s41579-020-0368-1

Vedantam G, Guay GG, Austria NE, Doktor SZ, Nichols BP. 1998. Characterization of mutations contributing to sulfathiazole resistance in *Escherichia coli*. *Antimicrob Agents Chemother* **42:** 88–93. doi:10.1128/AAC.42.1.88

Vicedomini R, Quince C, Darling AE, Chikhi R. 2021. Strainberry: automated strain separation in low-complexity metagenomes using long reads. *Nat Commun* **12:** 4485. doi:10.1038/s41467-021-24515-9

Warburton PJ, Amodeo N, Roberts AP. 2016. Mosaic tetracycline resistance genes encoding ribosomal protection proteins. *J Antimicrob Chemother* **71:** 3333–3339. doi:10.1093/jac/dkw304

Wick RR. 2019. Badread: simulation of error-prone long reads. *J Open Source Softw* **4:** 1316. doi:10.21105/joss.01316

Wick RR, Judd LM, Gorrie CL, Holt KE. 2017. Completing bacterial genome assemblies with multiplex MinION sequencing. *Microb Genom* **3:** e000132. doi:10.1099/mgen.0.000132

Wilm A, Aw PPK, Bertrand D, Yeo GHT, Ong SH, Wong CH, Khor CC, Petric R, Hibberd ML, Nagarajan N. 2012. Lofreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic Acids Res* **40:** 11189–11201. doi:10.1093/nar/gks918

Zagordi O, Bhattacharya A, Eriksson N, Beerenwinkel N. 2011. ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data. *BMC Bioinformatics* **12:** 119. doi:10.1186/1471-2105-12-119

Zhang T, Li H, Ma S, Cao J, Liao H, Huang Q, Chen W. 2023. The newest Oxford Nanopore R10.4.1 full-length 16S rRNA sequencing enables the accurate resolution of species-level microbial community profiling. *Appl Environ Microbiol* **89:** e0060523. doi:10.1128/aem.00605-23

Zhou Q, Ji F, Lin D, Liu X, Zhu Z, Ruan J. 2024. KSNP: a fast de Bruijn graph-based haplotyping tool approaching data-in time cost. *Nat Commun* **15:** 3126. doi:10.1038/s41467-024-47562-4

Received February 14, 2025; accepted in revised form September 9, 2025.