stRainy: assembly-based metagenomic strain phasing using long reads

Ekaterina Kazantseva[1,*], Ataberk Donmez[2,3,*], Mihai Pop[3,#], Mikhail Kolmogorov[2,#]

[1] ITMO University, St. Petersburg, Russia
[2] Center for Cancer Research, National Cancer Institute, National Institutes of Health, Bethesda, MD, USA
[3] Department of Computer Science, University of Maryland, College Park, MD

* Lead co-authors
[#] Corresponding authors: mpop@umd.edu and mikhail.kolmogorov@nih.gov

Abstract. Bacterial species in microbial communities are often represented by mixtures of strains. Variation in strain genomes may have important phenotypic effects, however strain-level deconvolution of microbial communities remains challenging. Short-read approaches can be used to detect small-scale variation between strains, but fail to phase these variants into contiguous haplotypes. Recent advances in long-read metagenomics resulted in complete de novo assemblies of various bacterial species. However, current assembly approaches often suppress strain-level variation, and instead produce species-level consensus representation. Strain variants are often unevenly distributed, and regions of high and low heterozygosity may interleave in the assembly graph, resulting in tangles. To address this, we developed an algorithm for metagenomic phasing and assembly called stRainy. Our approach takes a sequence graph as input, identifies graph regions that represent collapsed strains, phases them and represents the results in an expanded and simplified assembly graph. We benchmark stRainy using simulated data and mock metagenomic communities and show that it achieves strain-level deconvolution with high completeness and low error rates, compared to the other strain assembly and phasing approaches.

Introduction

Metagenomic sequencing of human microbiome and other complex microbial communities reveals extensive heterogeneity on sub-species levels (Zhao et al., 2019). Different strains of the same bacterial species often show distinct phenotypes, such as pathogenicity to humans (Kaper et al., 2004) or antimicrobial resistance (Schloissnig et al., 2013). Changes in strain composition may provide insights into microbiome evolution (Good et al, 2017), disease associations (Yan et al., 2020) and drug response (Zimmermann et al., 2019).

Reference-based quantification of bacterial species using Illumina shotgun metagenomic sequencing is a widely-used approach for profiling microbial communities (Kim et al., 2016, Wood & Langmead, 2019, Pierce et al., 2019). However, reference profiling can only detect strains that are already present in the databases. Alternatively, de novo strain profiling methods use short read mapping to detect variants and use the frequency information to quantify strain composition (Olm et al., 2021).

While the short-read reference-mapping approach can reliably detect small variants in strain genomes, it is difficult to phase these variants into longer haplotypes due to the limited length of the reads. In contrast, long-read sequencing technologies (such as Oxford Nanopore or PacBio) can connect variants that are tens of kilobases apart. For example, diploid human genomes can be phased into megabase-sized haplotypes (Martin et al., Edge et al., 2017, Shafin et al., 2021). New algorithms were also developed for polyploid genomes phasing (Schrinner et al., 2020).

The phasing methods designed for diploid or polyploid genomes are difficult to apply for metagenomic phasing (Nicholls et al., 2021), since the number of strains and their proportions in a metagenome are not known a priori. Recently, the authors of the Strainberry algorithm (Vicedomini et al., 2021) introduced an approach for phasing multiple bacterial strains, in which a pairwise phasing algorithm is used iteratively to separate one strain at a time. Below we illustrate that the pairwise phasing model may not account for uneven distribution of heterozygosity within multiple strains, leading to misassemblies.

Another method called MAGPhase was developed for long and accurate HiFi reads and can phase multiple strain haplotypes through the clustering of their characteristic SNPs (Bickhart et al, 2022). In another work, the authors of iGDA (Feng et al., 2021) made an improvement in error-prone long-read metagenomic SNP calling and phasing. Further, the recently introduced strainFlye pipeline (Fedarko et al., 2022) can detect and rare strain mutations in HiFi metagenomic sequencing. These tools however were primarily developed for local, rather than whole genome strain phasing.

Other studies explored the variant composition of diverse viral populations (Knyazev et al., 2021) using assembly (Jablonski et al., 2021) or long-read phasing (Warwick-Dugdale et al., 2019, Zhou et al., 2018). However, these methods can not be directly applied to more complex bacterial genomes.

Reference-based methods are limited to the identification of well-studied bacterial species that are present in the existing databases. Short-read metagenomic assembly can overcome the reference bias and improve de novo strain detection (Quince et al., 2021, Ghurye et al., 2019), but the assembled genomes are usually fragmented (Li et al. 2015, Nurk et al. 2017). Alternatively, long-read metagenomics can recover complete (or nearly-complete) microbial genomes from complex communities, such as human gut (Bertrand et al., 2019, Kim et al., 2022), wastewater (Dai et al., 2022), marine (Beaulaurier et al., 2020) or soil (Van Goethem et al., 2021). This capability was facilitated by the improvements in long-read metagenomic DNA extraction (Nicholls et al., 2019, Moss et al., 2020) and the rapid developments of long-read de novo assembly algorithms, such as Canu (Koren et al., 2017), Flye (Kolmogorov et al., 2019), Shasta (Shafin et al., 2020), hifiasm (Cheng et al., 2021), and others. To address the challenge of reduced sequencing depth, uneven species composition and long inter-species repeats, we recently developed a specialized long-read metagenomic assembler called metaFlye (Kolmogorov et al., 2020).

It has been shown that reconstruction of heterogeneous bacterial species (represented by multiple strains) is particularly challenging for long-read assemblers (Kolmogorov et al., 2020, Bickhart et al.,

2022, Meyer et al., 2022). Algorithms often fail to distinguish reads from closely-related strains, and instead produce a collapsed species-level consensus. The recently developed hifiasm-meta algorithm (Feng et al., 2022) improved the separation of closely-related strains by taking advantage of highly accurate Pacbio HiFi reads. However, hifiasm-meta currently is not applicable to sequencing technologies with higher error rates, such as Oxford Nanopore. Sequence graph-based phasing has been successfully applied to diploid genomes, but does not extend to the case of more than two haplotypes (Chin et al., 2016, Garg et al., 2020, Faure et al., 2021).

Here we present an algorithm for phasing and assembly of closely-related strains called stRainy. Our algorithm works with either a linear reference or a de novo assembly graph as input, and supports long reads with different error profiles. Because the strain variants are often unevenly distributed, regions of high and low heterozygosity may interleave in the assembly graph, which leads to tangles. stRainy identifies graph regions that represent collapsed strains, phases them and represents the results in an expanded and simplified assembly graph (Supplementary Figure 1). We benchmark stRainy using simulated and mock metagenomic datasets and show that it produces strain-level reconstructions with high completeness and low haplotype switch errors, and improves over the other tools for metagenomic phasing and assembly.

Methods

Assembly-based approach to the metagenomic phasing problem. A theoretical framework for the metagenomic phasing problem was recently described by Nicholls et al. (2021). Here we attempt to give a complementary assembly-based formulation. The definition of a bacterial strain is a subject of a separate debate (Olm et al., 2021, Bickhart et al., 2022), with some studies considering a single nucleotide change in a bacterial genome to be a characteristic of a new strain (or lineage). Here, we require strains to have a substantial number of characteristic variants, however the exact heterozygosity rate is a parameter that primarily depends on the read error profile, read length distribution and sequencing depth of the strains of interest. We call bacterial strains that satisfy these requirements for a particular dataset *resolvable*.

Traditionally, the phasing problem is defined for a linear reference genome, a set of heterozygous variants and reads that cover contiguous subsets of these variants. The goal of the phasing procedure is to assign the heterozygous genomic variants to their respective haplotypes. Although not biologically equivalent, resolvable strains act as haplotypes in the context of metagenomic phasing, and we use these terms interchangeably throughout the paper.

Alternatively, a complete and correct assembly of all haplotypes provides a solution to the phasing problem. In this work, we take an assembly-based approach to characterize metagenomic strains. Most de novo assembly algorithms utilize assembly graphs to generate the final contigs and represent remaining ambiguities. Here we use the definition of an assembly graph in which nodes contain assembled sequence, and edges representing adjacencies (and optional dovetail overlaps) between nodes. We call the entire sequence inside a graph node a *unitig*, as opposed to contig that can span multiple assembly graph nodes.

The goal of the final assembly is to ensure that: (i) each resolvable strain is fully covered by one or multiple contigs, and (ii) each assembled contig corresponds to exactly one resolvable strain. Current long-read assemblers often fail to separate reads from different strains, and as a result some assembled contigs may represent species level consensuses of multiple strains. We call contigs representing multiple strains *strain-collapsed*, and contigs that correspond to a single strain *strain-resolved*. Assembly graphs will often contain both strain-resolved and strain-collapsed unitigs which are interleaved, creating additional tangles (for example, bubbles or superbubbles).

Our stRainy algorithm takes as input a sequence graph containing strain-collapsed sequence and attempts to (i) phase each strain-collapsed unitig into multiple new unitigs that are strain-resolved and (ii) simplify the resulting assembly graph using the connectivity information from long reads. Because a linear reference genome could be viewed as an assembly graph with a single node, our approach is also applicable for phasing using a linear reference.

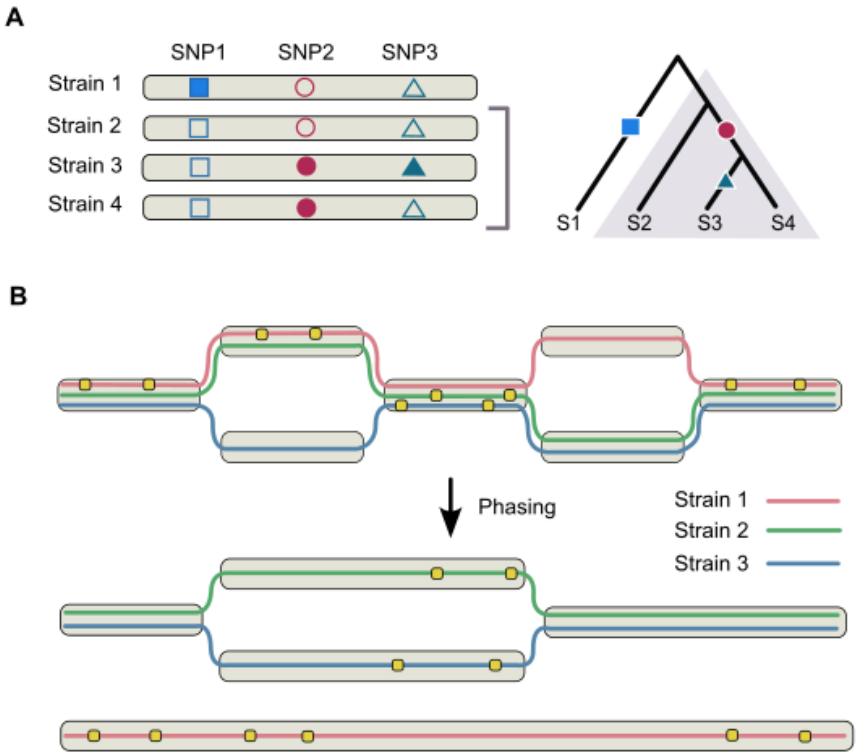


Figure 1. Metagenomic phasing problem. (A) Set of four closely-related strains and their corresponding phylogenetic tree. SNP positions are shown in different shapes, with hollow shapes representing reference genotype, and solid shapes representing mutant alleles. The subset of strains (S2, S3, S4) is highlighted. SNP1 is not informative for the highlighted subset, but other SNP positions are informative. (B) Simplified example of sequence graph phasing with three strains. Graph nodes are shown in gray, and three strain sequences are shown in colored lines. Strain-specific variants are shown with yellow rectangles. After phasing and graph simplification, most of the graph nodes are strain-resolved, but some nodes remain strain-collapsed.

Informative variants and graph representation of haplotype phasing. The goal of the unitig phasing stage is to phase each strain-collapsed unitig into the corresponding strain haplotypes. stRainy first processes each unitig of the input assembly graph independently. The connectivity of the original assembly graph is therefore not used at the unitig phasing stage. However, we illustrate below that the results of a single unitig phasing (of three or more haplotypes) are best represented in a newly formed *haplotype graph*, which will be embedded into the assembly graph at the next stages of the algorithm.

Let us assume that the unitig represents a set of collapsed and resolvable strains S of size $|S|$. Each strain $s \in S$ has the abundance of at least *minStrainFreq*. Strain haplotypes are distinguished using their characteristic variants. In this work we rely on single nucleotide polymorphisms (SNPs) since they typically can be called with much higher confidence (compared to indels) from long reads. A universal set of variants V could be computed by calling SNPs with frequencies at least *minStrainFreq* (was set to 0.1 in our benchmarks) from the unitig read alignment. Each variant v from V may be characteristic for one or multiple strains (including all of them in the case of an error in the unitig consensus sequence). Therefore, variant v splits the set of strains S into two subsets, S_v and $\overline{Sv}$, corresponding to strains containing and not containing the variant, respectively.

Given a subset of strains $T \subset S$, we call variant v *T-informative*, if v separates T into two non-zero subsets. For example, a variant contained in all strains from S is never informative. A variant that is

unique to a single strain *s* is always informative for subsets that include *s*. Which variants are informative for each strain subsets is determined by the evolutionary relationship between strains (Figure 1).

Characteristic strain variants are often distributed non-uniformly across a genome, creating regions of relatively high and low heterozygosity, with some regions having zero variable positions. A long non-heterozygous region is often difficult or impossible to phase. In diploid genomes, phased regions (or haplotype blocks) are interleaved with unphased regions of low heterozygosity. However, in the case of three or more haplotypes a linear definition may be insufficient. Figure 1 shows an example of three strains, one of which is more evolutionary distant, compared to two others. While the more distant strain can be fully separated, two other strains share long regions with no heterozygosity. For example, we identified 88 regions with no heterozygosity of length longer than 5kb between two *E. coli* strains, while there were 45 such regions shared by three *E. coli* strains and only 3 regions shared by four strains.

Forming fine strain clusters. We assume that reads are already aligned to the input graph unitigs, using minimap2 (Li, 2018) and SNPs with frequency at least *minStrainFreq* are identified. Reference-based phasing approaches typically call variants and then attempt to assign variants into phases using the connectivity information from reads. In human genome phasing, typical heterozygosity rate is relatively low (on average, 0.1%). Therefore most short read pairs (for which most phasing problems were originally formulated) rarely span even a single heterozygous variant, let alone two consecutive variants. In contrast, heterozygosity between collapsed bacterial strains is often higher, and most long reads span multiple strain variants. This opens the possibility for an assembly-based phasing problem formulation. Here, instead of assigning SNPs into their respective phases, we attempt to cluster long reads based on their strain of origin (or phase).

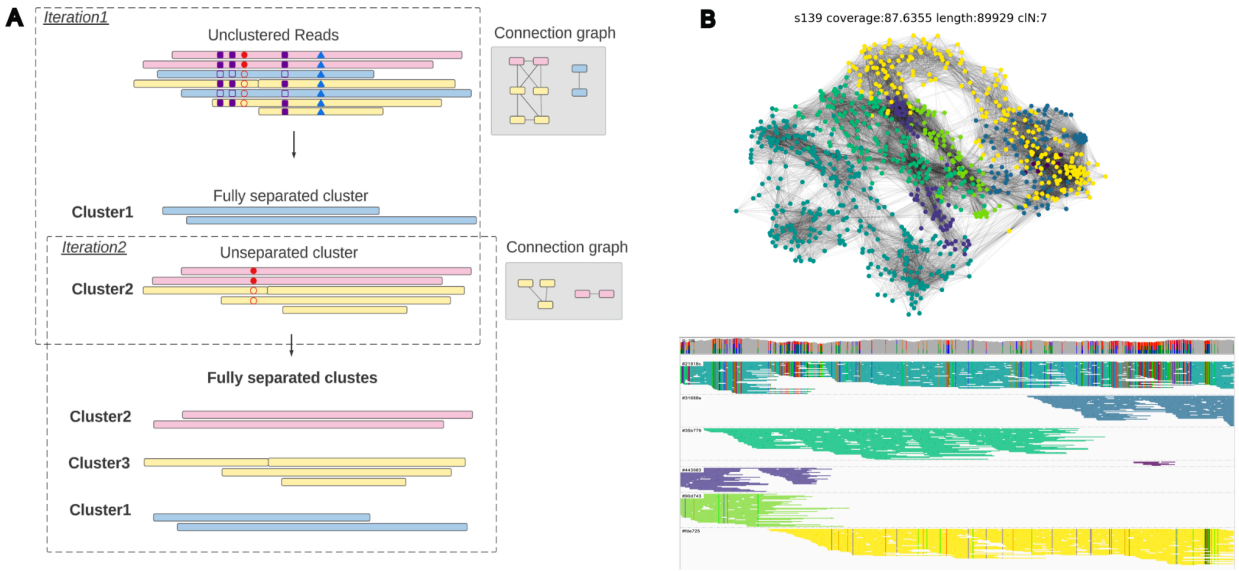


Figure 2. Unitig phasing. (A) Outline of the fine strain cluster generation algorithm. Given the reads and their variable positions as input, stRainy builds a connection graph which encodes the pairwise distances between reads. Afterwards, the community detection algorithm separates the graph into clusters. Next, we recompute the sets of informative SNPs, detect clusters that need further separation and iteratively repeat the community detection clustering. The final clusters represent reads originating from single strains. (B) An example of a unitig clustering using simulated HiFi reads from two strains as input.

Given a set of all reads *R* aligned to a unitig (either in full or partially), stRainy builds a *connection* graph as follows. Each node *n* corresponds to a read *r*, and two nodes n_1 , n_2 are connected with an undirected edge if the corresponding reads r_1 and r_2: (i) have a dovetail overlap of length at least *minOverlap* (typical value = 1000) (ii) overlap spans at least one informative SNP position and (iii) reads have at most

maxSnp non-matching informative SNP positions (in our benchmarks, *maxSNP* was set to zero for both HiFi and ONT). Note that the graph construction depends on the informative SNPs, but not the universal set of SNPs in a unitig. We will be iteratively using different sets of informative SNPs through the algorithm.

Intuitively, reads from the same strain of origin will form dense connected components on the connection graph (Figure 2). However, reads from different strains may also be connected. This is because even though we require read overlaps to span at least some SNPs, not all SNPs may be informative for all possible subsets of strains. This leads to false inter-strain connections.

Since the number of strains, their evolutionary relationship and the respective sets of informative SNPs are not known a priori, we use an iterative approach that gradually reduces the set of SNPs to perform fine-grade strain clustering. We begin with the initial connection graph (as described above) and the universal set of SNPs. At each iteration, stRainy partitions the graph using the community detection algorithm, separating the graph into dense connected components.

We assume that most of the clusters will contain reads from a single strain, but some clusters might be chimeric. Therefore, for each newly formed cluster, we reduce the set of informative SNPs according to the variant frequencies only from reads in this cluster. Afterwards, we remove connection graph edges that no longer span any informative SNP positions and repeat community detection clustering. The process is iteratively repeated until the set of informative SNP does not change substantially. In practice, more than two iterations are rarely needed: on a dataset consisting of five *E. coli* strains sequenced with HiFi reads, ~80% of clusters required a single iteration, and ~19% two iterations. We denote the resulting read grouping as *fine* clusters.

Reads for the regions of low heterozygosity may not span any informative SNPs and will correspond to orphaned nodes on the connection graph. Such singleton reads (or clusters of size less than *minCluster*) are separately clustered only based on their overlap coordinates, creating *blank* clusters.

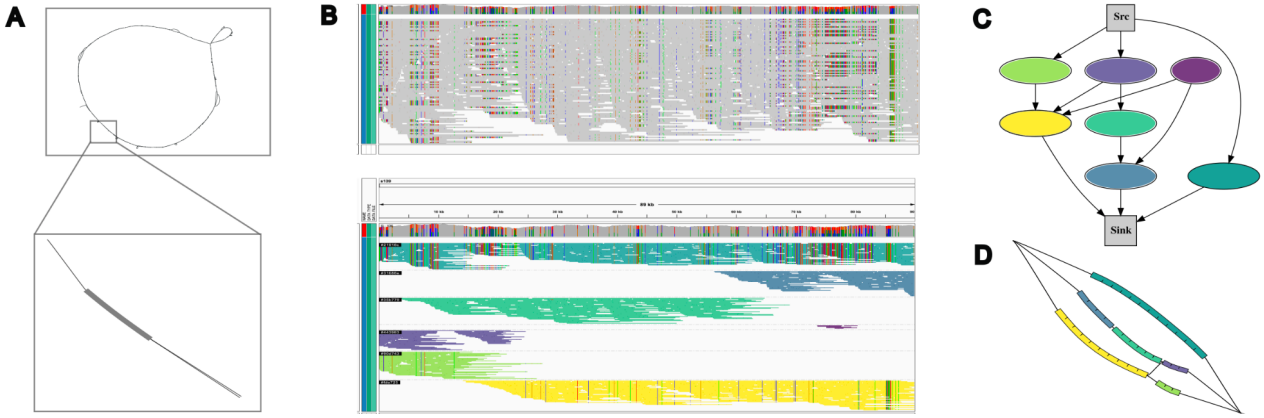


Figure 3. Strain graph construction and simplification. (a) A fragment of the input (unphased) graph visualized using Bandage. (b) Single strain-collapsed unitig. (c) Unphased read alignment against the strain-collapsed unitig, visualized using IGV. (d) Phased reads representing fins strain clusters. (e) Connection graph, with each node corresponding to a fine cluster + source and sink nodes. (f) The final resulting haplotype graph with strain-specific unitigs.

Strain graph assembly. Fine clusters designed to be highly accurate and contain reads from a single strain. However, a fine cluster may not cover an entire unitig because the regions of low heterozygosity result in reduced connectivity in the connection graph. As a result, the community detection algorithm may split the reads from the same strain into multiple fine clusters (Figure 3).

stRainy first computes the consensus of each fine cluster using the Flye polisher (Kolmogorov et al., 2019). Fine clusters from the same strain that are adjacent in a unitig will have overlaps. We then construct an overlap graph, where nodes correspond to consensus sequences of fine clusters. Two nodes are connected if the corresponding consensus sequences: (i) overlap by at least *minOverlap* bases; and (ii) have at most *maxSnp* differences. The graph also has source and sink nodes, corresponding to the unitig endpoints. Fine clusters that contain reads that reach the start or end of the unitig are connected to source or sink, respectively.

A fully resolved strain corresponds to a non-branching path from source to sink in the newly constructed overlap graph. However, blank clusters (that contain unphased reads from multiple strains) will result in forks. To recover strain haplotypes from the overlap graph, stRainy uses graph simplification procedures, similar to the ones used in de novo assembly. First, fine clusters that are fully contained (or nested) inside other clusters are removed. Second, small bubbles (formed by two alternative paths between two nodes) are transitively reduced.

Finally, non-branching paths are condensed into single unitigs. After simplification, each strain corresponds to a walk from the source to the sink. Short unitigs that do not belong to at least one such walk are likely results of read artifacts and are therefore removed from the graph. The graph after these simplifications is called strain graph and defines strain haplotypes, regions that are shared by multiple strains and their connectivity (Figure 3).

Assembly graph expansion and simplification. For each strain-collapsed unitig in the original assembly graph, stRainy performs the phasing procedure described above, resulting in a strain graph. Afterwards, each strain graph is embedded into the original assembly graph as follows. First, stRainy substitutes each original unitig with newly phased unitigs from the corresponding strain graph. After this operation, the number of strain-resolved sequences will substantially increase. However, some unitigs may be impossible to phase due to the lack of variation, and they will remain strain-collapsed. The goal of the final simplification step is to further extend the strain-resolved unitigs, but retain the connections with strain-collapsed sequences.

After adding strain-resolved unitigs to the graph, stRainy connects two newly formed unitigs if (i) they are connected in one of the strain graphs or (ii) if there are at least *minLink* reads that span both unitigs. The second condition is aimed at retaining the connectivity of the strain-collapsed nodes of the original assembly graph. In practice, we don't need to realign reads to test the condition, but instead reuse the information from the connection graph. Afterwards, stRainy performs additional simplification procedures on the updated assembly graph. If a node has two outgoing links, it is forming a *fork*. For every fork, we remove one of the outgoing connections if the resulting transformation does not result in new tips.

Results

Evaluating phasing and assembly approaches using simulated PacBio HiFi and Oxford Nanopore data. We first used simulated long-read sequencing data to benchmark stRainy along with metaFlye, Strainberry and hifiasm-meta (only on PacBio HiFi data). We used five *E. coli* strains (with pairwise ANI ranging from 93.3% to 97.7%) to simulate HiFi and ONT reads with different error rate distributions using Badread (Wick, 2019). For HiFi, we simulated 0.1% error rate and read length N50 = 5 kb. For ONT, we used the same length distribution and the mean error rate of 3% to mimic the recently improved ONT sequencing protocols (Liu et al., 2022). To directly compare the simulated results with the real metagenomic sequencing datasets below, we simulated reads from five *E. coli* strains from the Zymo D6331 synthetic community standard (*https://www.zymoresearch.com/collections/zymobiomics-microbial-community-standards*).

We simulated multiple series of datasets which included 2-5 *E. coli* strains to test the algorithms on metagenomic of various complexity. In addition, for both HiFi and ONT, we simulated two supersets of reads: a "uniform" dataset with read depth at 30x and "staggered" dataset with read depth decreasing in linear progression (e.g. 50x, 40x, 30x for a 3-strain dataset). Overall, there were four supersets of simulated reads, each containing datasets series with 2-5 strains (Supplementary Table 1)

stRainy takes a sequence graph in gfa format as input (which could be either de novo assembly graph or reference pangenome). We first evaluated stRainy using *E. coli* pangenome graphs constructed from the Zymo strains using minigraph (Li et al., 2020). The total sequence size of pan-genome graphs ranged from 5.6 Mb to 6.7 Mb, effectively representing a collapsed *E. coli* pangenome. Since Strainberry requires a linear reference, we used one of the Zymo *E. coli* strains (B1109) as input.

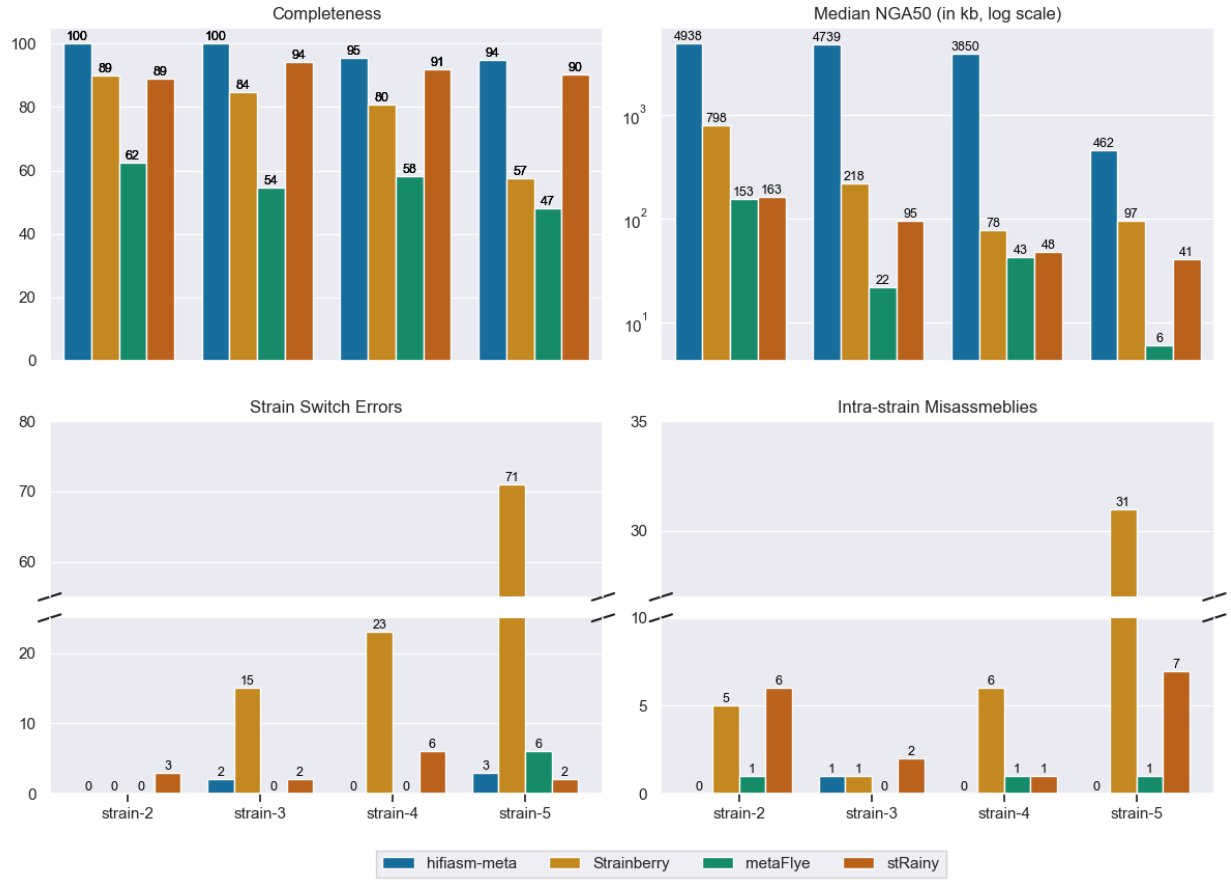


Figure 4. Benchmarking stRainy, Straimberry, hifiasm-meta and metaFlye using simulated HiFi sequencing data. Statistics are given for datasets with 2-5 strains with uniform read depth distribution. The statistics were computed using metaQUAST with the "--unique-mapping" option. A single assembled contig can only contribute to the coverage of one of the reference strains. Completeness reflects the total coverage of all reference strains. Strain switch error corresponds to the reported number of interspecies translocations, and intra-strain misassemblies correspond to all other misassmeblies reported by metaQUAST. Additional benchmarks using HiFi and ONT simulated datasets are shown on Supplementary Figures 2-4 and Supplementary Table 1.

To evaluate the accuracy and completeness of strain reconstruction, we then used metaQUAST (Mikheenko et al., 2016) that determines the strain of origin of each input contig based on the best alignment against all strain references. We used the "--unique-mapping" option to ensure a single contig (either strains-specific or strain-collapsed) can only contribute to the completeness of one of the reference strains. We estimated the number of switch errors (a chimeric connection between two distinct strains) as the number of interspecies translocations reported by metaQUAST.

On the uniform HiFi datasets with 2-5 strains, stRainy mean reference completeness was 91.3%, compared to 78.3% for Strainberry and for 56% metaFlye (Figure 4). De novo assembly using hifiasm-meta had completeness 97% - higher, but comparable to the stRainy result. On average, stRainy

and hifiasm-meta produced a very low number of switch errors on HiFi data (3.25 and 1.25, respectively), while the average number of switches in Strainberry was high (27.25). Assemblies of stRainy and Strainberry were more fragmented, compared to hifiasm-meta, reflected in a lower NGA50. Both stRainy and Strainberry had base-level sequence quality above QV35. As expected, reconstruction of staggered strain mixtures was more challenging for all algorithms, however the same performance trends were observed (Supplementary Figure 3).

On the uniform ONT simulated dataset, stRainy reconstructed on average 85% unique strain sequence, which was an improvement over Strainberry (79%) and metaFlye (38%) assemblies (Supplementary Figures 3-4). Because of the higher rate of errors in ONT reads, the number of interspecies switches slightly increased for all tools, compared to the HiFi dataset. stRainy remained more accurate than Strainberry (9.75 and 22.5 switch errors on average, respectively). Similarly, the number of mismatches and small indels in the was elevated for all methods, compared to the HiFi simulations. As expected compared to HiFi datasets, both stRainy and Strainberry had reduced base-level sequence quality, which remained above QV25.

We did not benchmark hifiasm-meta on the ONT data because the approach only supports highly accurate HiFi reads. We also attempted to run iGDA on the HiFi datasets, but the resulting contigs did not add any coverage to the collapsed strains. Running stRainy on most of the simulated HiFi and ONT datasets took less than 5 and 9 hours, respectively (wall-clock) on a standard laptop or a server node using 8 threads. We expect the performance to be further optimized with more effective parallelization strategies.

Simplifying de novo metagenome assembly graph using stRainy. Complex metagenomics communities often contain bacterial species that are lacking high quality reference sequences. In this case, de novo assembly could be used to reconstruct consensus representation of novel bacterial genomes. De novo metagenomic assembly graphs often contain either missing or chimeric connections. Yet another challenge is possible strain switching errors along graph unitigs, as the polishing algorithms may select arbitrary strain to represent the consensus sequence in case of disagreement. We thus benchmarked the ability of stRainy to work with metaFlye assembly graphs as input (assembled using the "--keep-haplotypes" option).

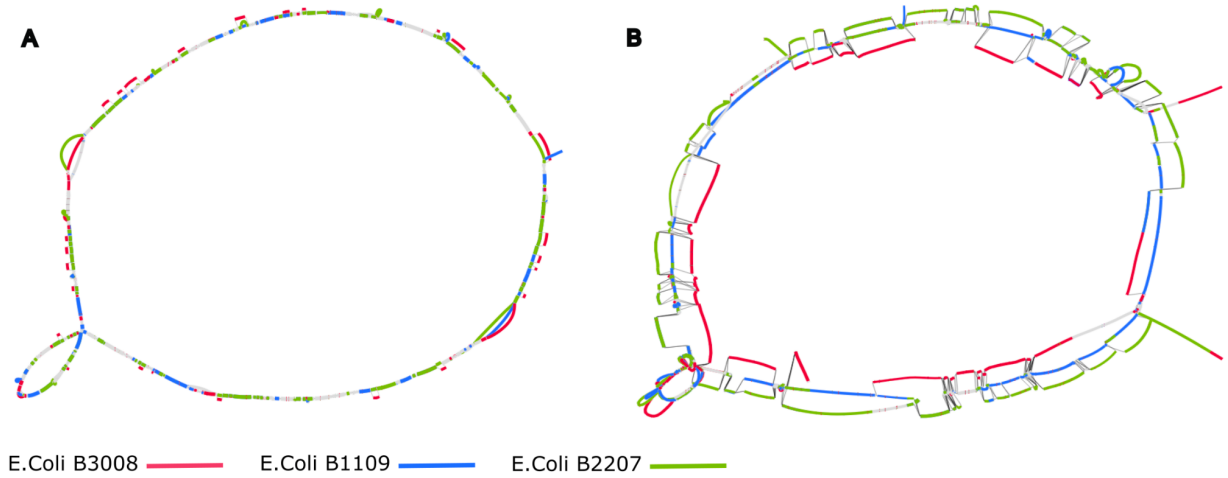


Figure 5. **Assembly graph transformation using stRainy** (Left) metaFlye assembly graph of 3 simulated E. coli strains. The total length of graph unitigs is 8.05 Mb, representing only 54.7% of the reference sequence. Each unitig is colored by the strain reference it has the best alignment to. (Right) Assembly graph after stRainy transformation, of total size 14.16 Mb, representing 92.7% of the reference sequence.

Figure 5 shows an example where stRainy transformed the metaFlye assembly of the simulated HiFi dataset, increasing strain completeness from 54.7% to 92.7%. Overall, using metaFlye assembly graphs as input for stRainy produced strain assemblies comparable to assemblies produced using reference graphs. In HiFi simulated datasets with 5 strains, completeness was slightly reduced (from 95.7% to 92.9%), and number of strain switches increased, but remained low (from 2 to 6). In the case of simulated ONT dataset, the changes were more noticeable: completeness decreased from 89.3% to 80.9%, and the number of strain switches increased from 7 to 16. This likely reflects overall lower quality of ONT assembly graphs, compared to HiFi graphs.

Evaluating using real HiFi and ONT sequencing of synthetic bacterial communities. Next, we evaluated stRainy and other tools using the real PacBio HiFi and ONT sequencing of synthetic bacterial communities. We downloaded publicly available HiFi (*https://github.com/PacificBiosciences/pb-metagenomics-tools/blob/master/docs/PacBio-Data.md*) and ONT (Liu et al., 2022) sequencing of the Zymo mock community D6331, which includes multiple bacterial species in single strains, and five *E. coli* strains. Since only the *E. coli* genome required phasing, we selected only the reads aligned to *E. coli* references for our analysis; for HiFi, we downsampled the resulting reads to ~30x read depth of each strain to be consistent with the simulated benchmark. We used the *E. coli* reference pan-genomes as input for stRainy.

On both HiFi and ONT datasets, stRainy strain reconstructions had substantially higher completeness (87.1% and 76.9%, respectively), compared to Strainberry (58.2% and 56.8%, respectively) and metaFlye (64.7% and 29.8%, respectively). stRainy also had the highest base-level quality (QV36), and made substantially less switch errors compared to Strainberry on both datasets. On the other hand, Strainberry reconstructions had higher NGA50, suggesting a more aggressive assembly approach.

Overall, assembly completeness of the real HiFi and ONT datasets was consistent with the simulated results. While both simulated and real HiFi datasets both had only a few switch errors, the number increased for the real ONT dataset, compared to the simulated mixture. This may be a result of a shorter read length of the real dataset (read N50 was ~2.5 kb, compared to 5 kb in the simulated dataset).

Method	Reference completeness	Reference identity	Strain switch errors	Intra-strain errors	Median NGA50 (kb)
HiFi Zymo D6331, total read length 0.7 Gb, read N50 6.1kb					
stRainy	**87.1%**	**QV36**	8	**12**	35.8
Strainberry	58.2%	QV27	27	67	**117.7**
metaFlye	64.7%	QV29	**6**	16	34.4
ONT Zymo D6331, total read length, 1.2 Gb, read N50 2.5 kb					
stRainy	**76.9%**	**QV27**	39	47	9.3
Strainberry	56.8%	QV26	60	26	69.0
metaFlye	29.8%	QV25	**10**	**9**	-

Table 1. Benchmarking stRainy, metaFlye and Strainberry on real HiFi and ONT sequencing of mock microbial communities. Statistics were computed using metaQUAST with "--unique-mapping" option. The best values in each category are highlighted in bold.

Discussion

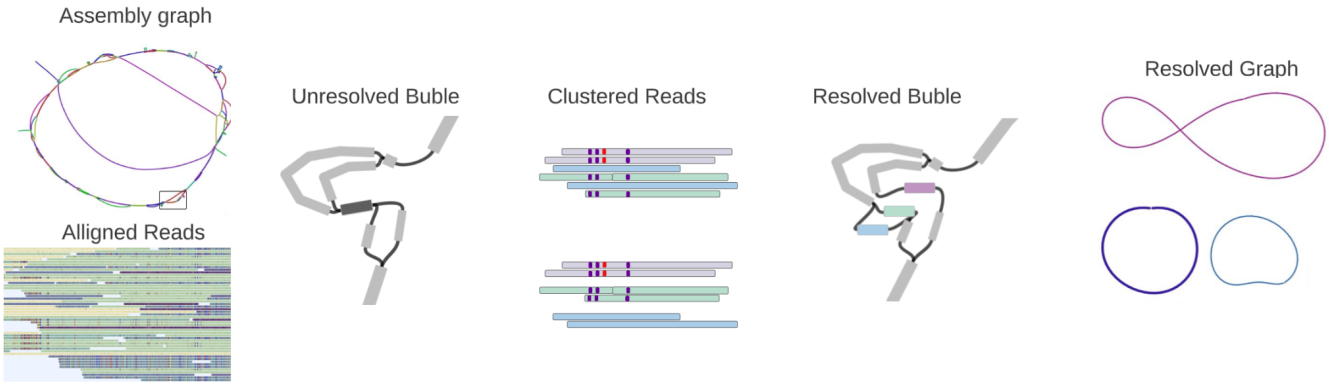
In this work we presented an algorithmic framework for long-read assembly-based phasing of bacterial strains. In contrast to previous approaches, our algorithm can operate on sequence graphs and therefore improve de novo assemblies or take advantage of pangenome reference graphs. We also described that the haplotype graph is a natural solution of the multi-allele phasing problem, which may not always have linear representation of phased blocks if heterozygous positions are unevenly distributed in strain genomes. We implemented the framework into a stRainy tool, and tested it on various simulated and real datasets with different strain composition and abundance. We showed that stRainy substantially improved over the previous metagenomic phasing approach, and the algorithmic framework is general enough to be applied to both PacBio HiFi and Oxford Nanopore technologies with different error patterns.
The multi-allelic phasing approach could also be adopted to other applications, such as phasing of segmental duplication copies in the human genome and phasing of polyploid genomes.

In our experiments, hifiasm-meta produced the best HiFi assemblies, but the algorithm is difficult to adapt to ONT sequencing with an elevated number of errors and their nonrandom distribution. In addition, the current stRainy algorithm is relatively conservative, and implicitly requires support of multiple reads to phase adjacent variants. We expect more improvements in the stRainy graph simplification procedures are possible, which in turn will result in more contiguous strain haplotypes.

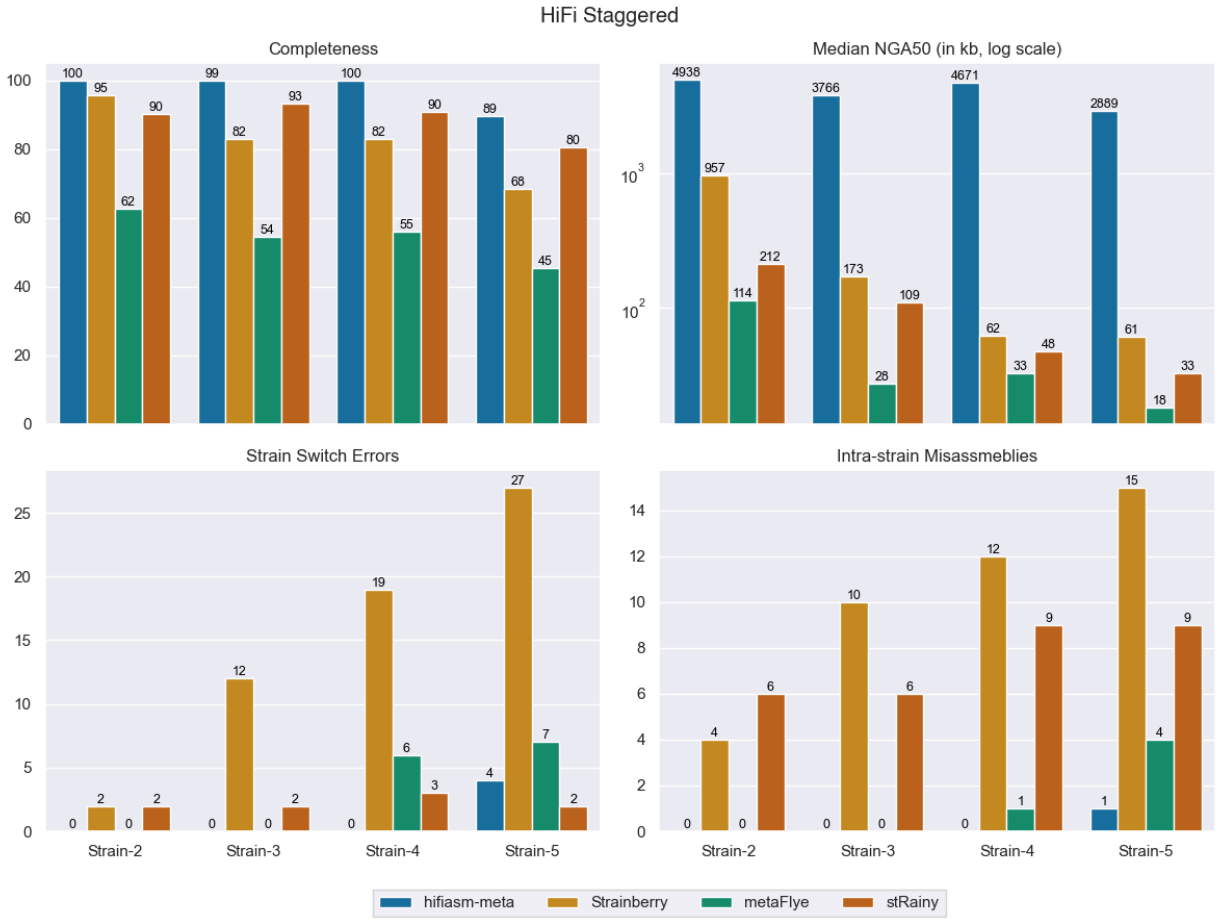
Using de novo metagenomic assembly graphs proved to be more challenging for phasing, compared to the pangenome reference input. We speculate that this is primarily due to the imperfections of the assembly graphs, which may contain missing and chimeric connections, as well as phase switches. Using de novo assembly graphs is nevertheless an important method for analyzing bacterial species with limited representation in genome databases. We are planning to improve the integration of stRainy with metaFlye and other metagenomic assemblers via better mapping of reads to graphs and structural error detection. In this work we tested stRainy on relatively small metagenomes, and the algorithms to large and complex bacterial communities is another avenue for improvement. It should be feasible to initially separate bacterial species on metagenome assembly graphs as they will form separate connected components.

Code availability. stRainy is freely available at: https://github.com/katerinakazantseva/stRainy

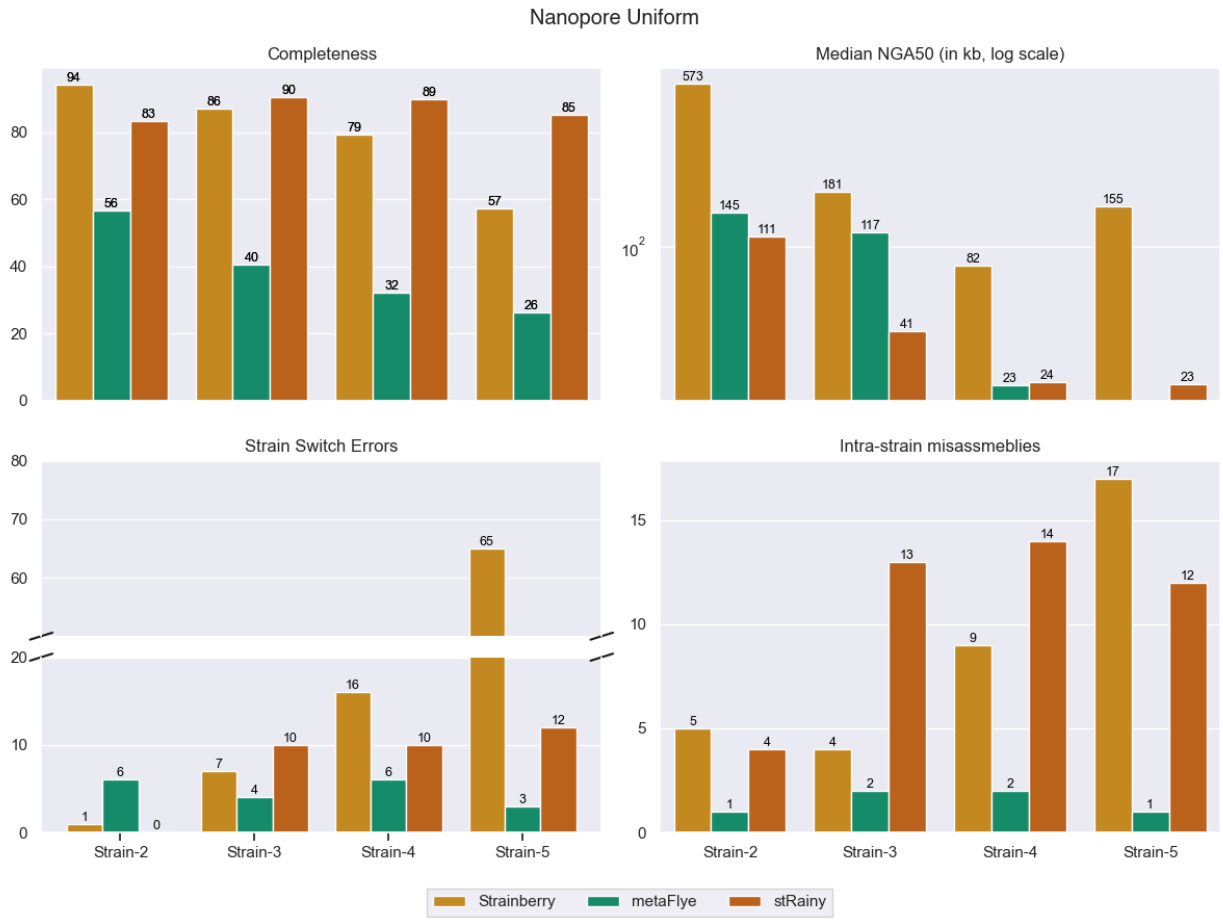
Supplementary Figures



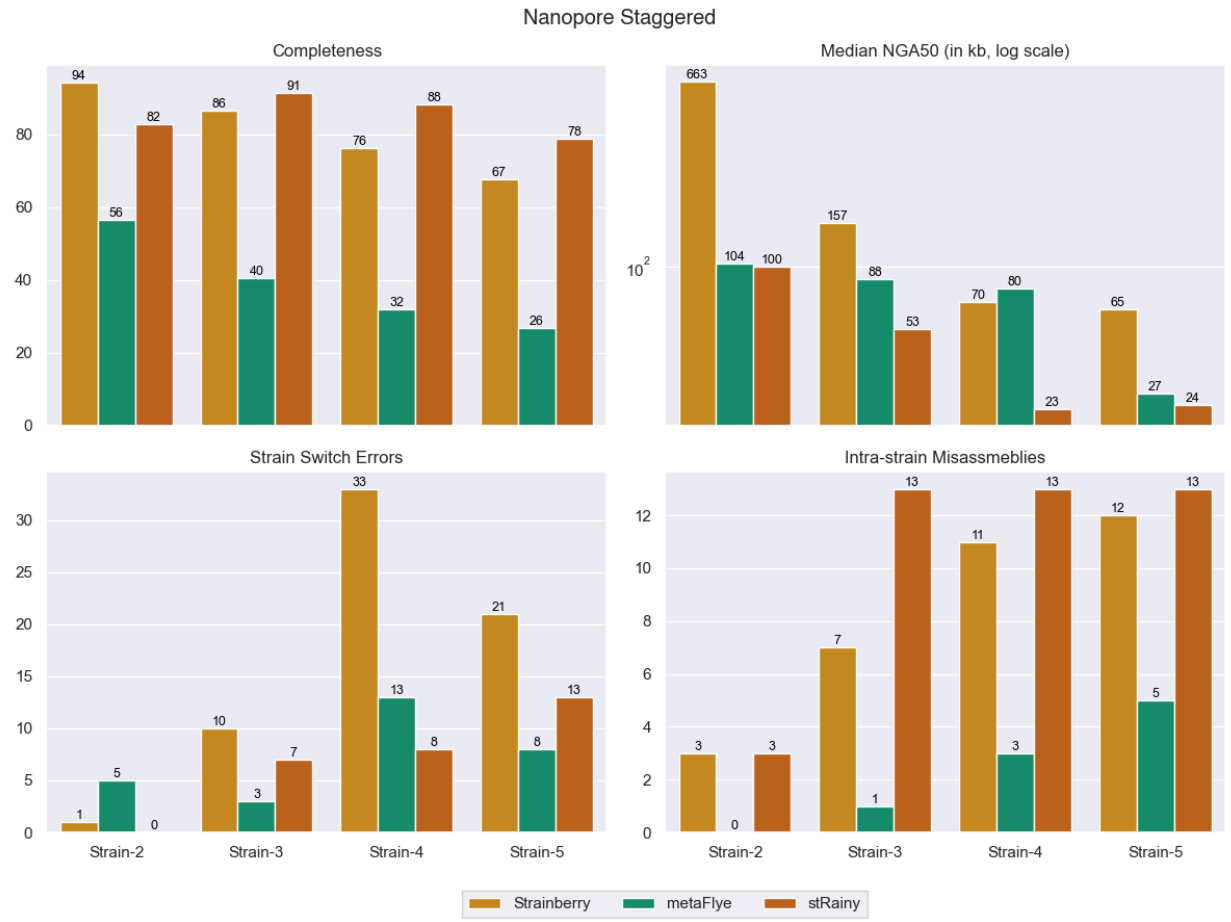
Supplementary Figure 1. High-level overview of the stRainy pipeline.



Supplementary Figure 2. Benchmarking stRainy, Straimberry, hifiasm-meta and metaFlye using simulated HiFi (staggered) sequencing data.



Supplementary Figure 3. Benchmarking stRainy, Straimberry, hifiasm-meta and metaFlye using simulated ONT (uniform) sequencing data.



Supplementary Figure 4. Benchmarking stRainy, Straimberry, hifiasm-meta and metaFlye using simulated ONT (staggered) sequencing data.

References

Ghurye, J., Treangen, T., Fedarko, M. et al. MetaCarvel: linking assembly graph motifs to biological variants. Genome Biol 20, 174 (2019). https://doi.org/10.1186/s13059-019-1791-3

Nurk S, Meleshko D, Korobeynikov A, Pevzner PA. metaSPAdes: a new versatile metagenomic assembler. Genome Res. 2017 May;27(5):824-834. doi: 10.1101/gr.213959.116. Epub 2017 Mar 15. PMID: 28298430; PMCID: PMC5411777.

Dinghua Li, Chi-Man Liu, Ruibang Luo, Kunihiko Sadakane, Tak-Wah Lam, MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph, Bioinformatics, Volume 31, Issue 10, 15 May 2015

Moss, Eli L., Dylan G. Maghini, and Ami S. Bhatt. "Complete, closed bacterial genomes from microbiomes using nanopore sequencing." Nature biotechnology 38.6 (2020): 701-707.

Koren, Sergey, Brian P. Walenz, Konstantin Berlin, Jason R. Miller, Nicholas H. Bergman, and Adam M. Phillippy. "Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation." Genome research 27, no. 5 (2017): 722-736.

Kolmogorov, Mikhail, et al. "Assembly of long, error-prone reads using repeat graphs." Nature biotechnology 37.5 (2019): 540-546.

Shafin, Kishwar, Trevor Pesout, Ryan Lorig-Roach, Marina Haukness, Hugh E. Olsen, Colleen Bosworth, Joel Armstrong et al. "Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes." Nature biotechnology 38, no. 9 (2020): 1044-1053.

Cheng, Haoyu, Gregory T. Concepcion, Xiaowen Feng, Haowen Zhang, and Heng Li. "Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm." Nature methods 18, no. 2 (2021): 170-175.

Bertrand, Denis, Jim Shaw, Manesh Kalathiyappan, Amanda Hui Qi Ng, M. Senthil Kumar, Chenhao Li, Mirta Dvornicic et al. "Hybrid metagenomic assembly enables high-resolution analysis of resistance determinants and mobile elements in human microbiomes." Nature biotechnology 37, no. 8 (2019): 937-944.

Dai, Dongjuan, Connor Brown, Helmut Bürgmann, D. G. Larsson, Indumathi Nambi, Tong Zhang, Carl-Fredrik Flach, Amy Pruden, and Peter J. Vikesland. "Long-read metagenomic sequencing reveals shifts in associations of antibiotic resistance genes with mobile genetic elements from sewage to activated sludge." Microbiome 10, no. 1 (2022): 1-16.

Beaulaurier, John, Elaine Luo, John M. Eppley, Paul Den Uyl, Xiaoguang Dai, Andrew Burger, Daniel J. Turner et al. "Assembly-free single-molecule sequencing recovers complete virus genomes from natural microbial communities." Genome Research 30, no. 3 (2020): 437-446.

Van Goethem, Marc W., Andrew R. Osborn, Benjamin P. Bowen, Peter F. Andeer, Tami L. Swenson, Alicia Clum, Robert Riley et al. "Long-read metagenomics of soil communities reveals phylum-specific secondary metabolite dynamics." Communications biology 4, no. 1 (2021): 1-10.

Kolmogorov, Mikhail, Derek M. Bickhart, Bahar Behsaz, Alexey Gurevich, Mikhail Rayko, Sung Bong Shin, Kristen Kuhn et al. "metaFlye: scalable long-read metagenome assembly using repeat graphs." Nature Methods 17, no. 11 (2020): 1103-1110.

Bickhart, Derek M., Mikhail Kolmogorov, Elizabeth Tseng, Daniel M. Portik, Anton Korobeynikov, Ivan Tolstoganov, Gherman Uritskiy et al. "Generating lineage-resolved, complete metagenome-assembled genomes from complex microbial communities." Nature biotechnology 40, no. 5 (2022): 711-719.

Feng, Xiaowen, Haoyu Cheng, Daniel Portik, and Heng Li. "Metagenome assembly of high-fidelity long reads with hifiasm-meta." Nature Methods (2022): 1-4.

Feng, Zhixing, Jose C. Clemente, Brandon Wong, and Eric E. Schadt. "Detecting and phasing minor single-nucleotide variants from long-read sequencing data." Nature communications 12, no. 1 (2021): 1-13.

Nicholls, Samuel M., Wayne Aubrey, Kurt De Grave, Leander Schietgat, Christopher J. Creevey, and Amanda Clare. "On the complexity of haplotyping a microbial community." Bioinformatics 37, no. 10 (2021): 1360-1366.

Vicedomini, Riccardo, Christopher Quince, Aaron E. Darling, and Rayan Chikhi. "Strainberry: automated strain separation in low-complexity metagenomes using long reads." Nature Communications 12, no. 1 (2021): 1-14.

Raghavan, Usha Nandini, Réka Albert, and Soundar Kumara. "Near linear time algorithm to detect community structures in large-scale networks." Physical review E 76, no. 3 (2007): 036106.

Li, Heng. "Minimap2: pairwise alignment for nucleotide sequences." Bioinformatics 34.18 (2018): 3094-3100.

Danecek, Petr, et al. "Twelve years of SAMtools and BCFtools." Gigascience 10.2 (2021): giab008.

Wick, Ryan R., Mark B. Schultz, Justin Zobel, and Kathryn E. Holt. "Bandage: interactive visualization of de novo genome assemblies." Bioinformatics 31, no. 20 (2015): 3350-3352.

Rautiainen, Mikko, and Tobias Marschall. "GraphAligner: rapid and versatile sequence-to-graph alignment." Genome biology 21, no. 1 (2020): 1-28.

Kim, C.Y., Ma, J. & Lee, I. HiFi metagenomic sequencing enables assembly of accurate and complete genomes from human gut microbiota. Nat Commun 13, 6367 (2022)

Samuel M Nicholls, Joshua C Quick, Shuiquan Tang, Nicholas J Loman, Ultra-deep, long-read nanopore sequencing of mock microbial community standards, GigaScience, Volume 8, Issue 5, May 2019

Schloissnig, S., Arumugam, M., Sunagawa, S. et al. Genomic variation landscape of the human gut microbiome. Nature 493, 45–50 (2013)

Zhao S, Lieberman TD, Poyet M, Kauffman KM, Gibbons SM, Groussin M, Xavier RJ, Alm EJ. Adaptive Evolution within Gut Microbiomes of Healthy People. Cell Host Microbe. 2019

Kaper JB, Nataro JP, Mobley HL. Pathogenic Escherichia coli. Nat Rev Microbiol. 2004

Good, B., McDonald, M., Barrick, J. et al. The dynamics of molecular evolution over 60,000 generations. Nature 551, 45–50 (2017)

Yan, Y., Nguyen, L.H., Franzosa, E.A. et al. Strain-level epidemiology of microbial communities and the human microbiome. Genome Med 12, 71 (2020)

Zimmermann, M., Zimmermann-Kogadeeva, M., Wegmann, R. et al. Mapping human microbiome drug metabolism by gut bacteria and their genes. Nature 570, 462–467 (2019).

Kim D, Song L, Breitwieser FP, Salzberg SL. Centrifuge: rapid and sensitive classification of metagenomic sequences. Genome Res. 2016

Wood DE, Lu J, Langmead B. Improved metagenomic analysis with Kraken 2. Genome Biol. 2019

Pierce NT, Irber L, Reiter T, Brooks P, Brown CT. Large-scale sequence comparisons with sourmash. F1000Res. 2019

Olm, M.R., Crits-Christoph, A., Bouma-Gregson, K. et al. inStrain profiles population microdiversity from metagenomic data and sensitively detects shared microbial strains. Nat Biotechnol 39, 727–736 (2021)

Marcel Martin, Murray Patterson, Shilpa Garg, Sarah O Fischer, Nadia Pisanti, Gunnar W Klau, Alexander Schöenhuth, Tobias Marschall. “WhatsHap: fast and accurate read-based phasing” bioRxiv 2016 doi: https://doi.org/10.1101/085050

Edge P, Bafna V, Bansal V. HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies. Genome Res. 2017

Shafin K, Pesout T, Chang PC, Nattestad M, Kolesnikov A, Goel S, Baid G, Kolmogorov M, Eizenga JM, Miga KH, Carnevali P, Jain M, Carroll A, Paten B. Haplotype-aware variant calling with PEPPER-Margin-DeepVariant enables high accuracy in nanopore long-reads. Nat Methods. 2021

Schrinner, S.D., Mari, R.S., Ebler, J. et al. Haplotype threading: accurate polyploid phasing from long reads. Genome Biol 21, 252 (2020). https://doi.org/10.1186/s13059-020-02158-1

Knyazev S, Hughes L, Skums P, Zelikovsky A. Epidemiological data analysis of viral quasispecies in the next-generation sequencing era. Brief Bioinform. 2021 Jan 18;22(1):96-108. doi: 10.1093/bib/bbaa101. PMID: 32568371; PMCID: PMC8485218.

Jablonski, Kim Philipp, and Niko Beerenwinkel. "Computational Methods for Viral Quasispecies Assembly." In Virus Bioinformatics, pp. 51-64. Chapman and Hall/CRC, 2021.

Warwick-Dugdale, J., Solonenko, N., Moore, K., Chittick, L., Gregory, A.C., Allen, M.J., Sullivan, M.B. and Temperton, B., 2019. Long-read viral metagenomics captures abundant and microdiverse viral populations and their niche-defining genomic islands. PeerJ, 7, p.e6800.

Zhou, Z., Luhmann, N., Alikhan, N. F., Quince, C., & Achtman, M. (2018, April). Accurate reconstruction of microbial strains from metagenomic sequencing using representative reference genomes. In International Conference on Research in Computational Molecular Biology (pp. 225-240). Springer, Cham.

Quince, C., Nurk, S., Raguideau, S. et al. STRONG: metagenomics strain resolution on assembly graphs. Genome Biol 22, 214 (2021). https://doi.org/10.1186/s13059-021-02419-7

Chin CS, Peluso P, Sedlazeck FJ, Nattestad M, Concepcion GT, Clum A, Dunn C, O'Malley R, Figueroa-Balderas R, Morales-Cruz A, Cramer GR, Delledonne M, Luo C, Ecker JR, Cantu D, Rank DR, Schatz MC. Phased diploid genome assembly with single-molecule real-time sequencing. Nat Methods. 2016 Dec;13(12):1050-1054. doi: 10.1038/nmeth.4035. Epub 2016 Oct 17. PMID: 27749838; PMCID: PMC5503144.

Roland Faure, Nadège Guiglielmoni, Jean-François Flot. GraphUnzip: unzipping assembly graphs with long reads and Hi-C. bioRxiv 2021, doi: https://doi.org/10.1101/2021.01.29.428779

Garg S, Aach J, Li H, Sebenius I, Durbin R, Church G. A haplotype-aware de novo assembly of related individuals using pedigree sequence graph. Bioinformatics. 2020;36(8):2385-2392. doi:10.1093/bioinformatics/btz942

Liu, Lei, Yu Yang, Yu Deng, and Tong Zhang. "Nanopore long-read-only metagenomics enables complete and high-quality genome reconstruction from mock and complex metagenomes." Microbiome 10, no. 1 (2022): 209.

Meyer, Fernando, Adrian Fritz, Zhi-Luo Deng, David Koslicki, Till Robin Lesker, Alexey Gurevich, Gary Robertson et al. "Critical assessment of metagenome interpretation: the second round of challenges." Nature methods 19, no. 4 (2022): 429-440.

Fedarko, Marcus W., Mikhail Kolmogorov, and Pavel A. Pevzner. "Analyzing rare mutations in metagenomes assembled using long and accurate reads." Genome Research 32, no. 11-12 (2022): 2119-2133.

Wick, Ryan R. "Badread: simulation of error-prone long reads." Journal of Open Source Software 4, no. 36 (2019): 1316.

Li, Heng, Xiaowen Feng, and Chong Chu. "The design and construction of reference pangenome graphs with minigraph." Genome biology 21 (2020): 1-19.

Mikheenko, Alla, Vladislav Saveliev, and Alexey Gurevich. "MetaQUAST: evaluation of metagenome assemblies." Bioinformatics 32, no. 7 (2016): 1088-1090.