Published in final edited form as:
Nat Methods. 2020 November ; 17(11): 1103–1110. doi:10.1038/s41592-020-00971-x.

metaFlye: scalable long-read metagenome assembly using repeat graphs

Mikhail Kolmogorov[1], **Derek M. Bickhart**[3], **Bahar Behsaz**[4], **Alexey Gurevich**[5], **Mikhail Rayko**[5], **Sung Bong Shin**[6], **Kristen Kuhn**[6], **Jeffrey Yuan**[4], **Evgeny Polevikov**[5,7], **Timothy P. L. Smith**[6], **Pavel A. Pevzner**[1,2]

[1]Department of Computer Science and Engineering, University of California, San Diego, CA, 92093, USA

[2]Center for Microbiome Innovation, University of California, San Diego, CA, 92093

[3]Cell Wall Biology and Utilization Laboratory, Dairy Forage Research Center, USDA, Madison, WI, 53706, USA

[4]Graduate Program in Bioinformatics and System Biology, University of California, San Diego, CA, 92093, USA

[5]Center for Algorithmic Biotechnology, St. Petersburg State University, Russia

[6]USDA-ARS U.S. Meat Animal Research Center, Clay Center, NE, 68933, USA

[7]Bioinformatics Institute, St. Petersburg, Russia

Abstract

Long-read sequencing technologies have substantially improved the assemblies of many isolate bacterial genomes as compared to the fragmented short-read assemblies. However, assembling complex metagenomic datasets remains difficult even for state-of-the-art long-read assemblers. Here we present the metaFlye algorithm that addresses important long-read metagenomic assembly challenges, such as uneven bacterial composition and intra-species heterogeneity. First, we benchmarked metaFlye using simulated and mock bacterial communities, and show

Corresponding author: P.A.P, ppevzner@ucsd.edu.

Author contributions. M.K., J.Y, and P.P. developed the metaFlye concept. M.K. implemented and maintained metaFlye. E.P. implemented the short plasmid analysis module. D.B., S.B.S., K.K., and T.S. performed the sheep gut sequencing. M.K., D.B., A.G., M.R. benchmarked metaFlye, and analyzed results. A.G. and M.K. performed the analysis of synthetic datasets. M.R. analyzed plasmid and virus content. B.B. performed the analysis of biosynthetic gene clusters. M.K., D.B., B.B., A.G., M.R., T.S., and P.P. edited the manuscript. P.P. supervised the project. All authors read and approved the manuscript.

Code availability.

metaFlye is freely available as a part of the Flye package at: https://github.com/fenderglass/Flye. The pbclip tool for PacBio subread splitting is available from https://github.com/fenderglass/pbclip.

Life Sciences Reporting Summary is available for this manuscript.

Ethics declaration. The sheep fecal sample was collected postmortem from the lower colon of a Katahdin breed wether (sheep, *Ovis aires*) that was raised in a ranch pasture setting. The animal died naturally and post-mortem was diagnosed with combined *Strongyloides* and coccidial infection. No ethical guidance was required as pre-mortem, the animal was cared for under approved guidelines for the handling of farm animals by the standard operating procedures of the Institutional Care and Use Committee. The animal did not display observable disease until shortly before death. All sheep in the flock that die are routinely necropsied to determine the cause of death to further improve handling guidelines, and the fecal sample used for the present study was a portion of that collected to generate the diagnosis.

Competing interests. The authors declare no competing interests.

that it consistently produces assemblies with better completeness and contiguity as compared to state-of-the-art long-read assemblers. Second, we performed long-read sequencing of the sheep microbiome and applied metaFlye to reconstruct 63 complete or nearly-complete bacterial genomes within single contigs. Finally, we show that the long-read assembly of the human microbiomes enables the discovery of novel biosynthetic gene clusters that encode biomedically important natural products.

Introduction

Bacterial genome assemblies produced from long Single Molecule Sequencing reads (generated using Pacific Biosciences or Oxford Nanopore technologies) are substantially more contiguous compared to short-read assemblies [1, 2]. In contrast, early long-read metagenomic studies reported lower yields and reduced read lengths compared to isolate bacterial assemblies, which made it difficult to generate high-quality assemblies and suggested that sample preparation protocols have to be optimized to utilize long reads in metagenomic studies [3, 4]. However, the recent improvements in high molecular weight DNA extraction techniques have enabled the sequencing of complex metagenomes with deep coverage and increased read lengths [5, 6, 7, 8]. Although these improved protocols have already been used for analyzing complex bacterial communities [9, 10 11, 12], there is still no specialized long-read metagenomic assembler. Indeed, although some long-read assemblers [13, 14, 15, 16, 17] have been applied to metagenomic datasets, none of them were designed to handle the specific challenges of metagenome assembly. This is unfortunate since long-read metagenomic assemblies have the potential to greatly improve upon the contiguity of short-read assemblies and address their inherent limitations, such as strain resolution [18], detection of horizontal gene transfer [19], difficulties in the search for new candidate phyla [20], sequencing of novel plasmids and viruses [21], and search for biomedically important biosynthetic gene clusters [22]. Long-read metagenomic assemblers are also important for improving the performance of hybrid assemblers that combine short and long reads [6, 8].

Metagenomic assembly presents additional computational challenges compared to the assembly of isolates due to the highly non-uniform coverage of the composing species, the presence of long intra-genomic and inter-genomic repeats [23, 24], and inter- and intra-species heterogeneity [25, 26]. We recently developed a fast long-read genome assembler, Flye, and showed that it produces accurate and contiguous assemblies [16]. Here we describe the metaFlye algorithm for long-read metagenome assembly, benchmark it using a diverse set of simulated, mock, and real bacterial communities, and demonstrate that it improves over state-of-the-art long-read assemblers Canu [15], FALCON [13], miniasm [14], OPERA-MS [6], and wtdbg2 [17].

Results

Assembly of species with highly uneven coverage.

The Flye algorithm (designed for single genome assembly) first attempts to approximate the set of *genomic* k-mers (k-mers that appear in the genome) by selecting *solid* k-mers (high-frequency *k*-mers in the read-set). It further uses solid *k*-mers to efficiently detect overlapping reads, and builds *disjointigs* [16]. However, in a metagenome setting, this approach would favor high-abundance species, while low-abundance species will have a reduced number of solid *k*-mers (if any), and thus will fail to be assembled. Here we introduce a different approach to solid *k*-mer selection, which combines global *k*-mer counting with analyzing local *k*-mer distributions (Methods). In addition, we describe an algorithm for the detection of repeat edges in the metagenome assembly graphs, which is robust to highly non-uniform distribution of read coverage (Figure 1a; Methods).

Assembling multiple closely-related bacterial genomes.

Another important metagenome assembly challenge is the presence of species with highly similar genomes in the sample. The related strains and species often contain shared conserved sequences as well as regions that are unique for each genome. We refer to each genome within a group of related species/strains as a *strain genome*. The shared and strain-specific regions generate *bubble structures* [27,28] in the repeat graph: *simple bubbles* in the cases of two strains (Figure 1b) and *superbubbles* in the case of more than two strains (Figure 1c). Moreover, some strain genomes may share repetitive sequences with the other unrelated genomes, which results in *roundabouts* (Figure 1d). Similarly to haplotype-aware assembly [29], these strain-induced subgraphs in the repeat graphs need to be detected and simplified to produce accurate and contiguous metagenomic assemblies [26]. The Methods section describes how metaFlye detects and simplifies strain-induced subgraphs. In addition to the standard *strain-suppression* mode, metaFlye also has a *strain-resolution* mode that we refer to as metaFlye$_{\text{strain}}$.

Benchmarking using simulated metagenomic datasets.

Long-read assemblers often generate complete assemblies for many genomes in mock community datasets, but fragmented assemblies of more complex real metagenomes. Ideally, one could benchmark assembly algorithms using a realistic complex mock dataset with known reference genomes, however, no such dataset is currently available. We thus simulated two complex bacterial communities with 64 and 181 genomes and benchmarked metaFlye, Canu, miniasm, and wtdbg2 on these two datasets that we refer to as SYNTH64 and SYNTH181, respectively (Supplementary Notes 1-2, Supplementary Tables 1-2). Here we summarize the benchmarking results on the SYNTH181 dataset generated based on a realistic bacterial community, originally described by the Critical Assessment of Metagenome Interpretation (CAMI) consortium [30].

First, we selected 181 complete bacterial reference genomes that were available for the CAMI_I_TOY_MEDIUM community (Supplementary Note 1). The analysis of these genomes using fastANI [31] showed that there were 33 genomes with closely-related strains (ANI >95%) and 22 genomes with closely-related species (ANI 85–95%), resulting in 55 genomes that are particularly challenging for long-read assemblers. We simulated 26 Gb of PacBio reads using Badread [32], following the abundance distributions from the original dataset (mode D1). The read coverage of each genome was varying from 0.01x to 497x. There were 91 out of 181 genomes with coverage above 5x.

metaFlye showed a substantial improvement over other assemblers in both contiguity and reference coverage of separate genomes on the SYNTH181 dataset (Figure 2), with improvements becoming more apparent for difficult-to-assemble genomes (characterized by low mean NGA50 and coverage among all assemblers). metaFlye/ metaFlye$_{strain}$ produced the assemblies with a higher total metagenome reference coverage (54.8%/54.1%), followed by Canu (43.1%), miniasm (42.9%), wtdbg2 (42.7%) and Flye (24.3%). metaFlye and metaFlye$_{strain}$ assembled over 90% of the total length of the 92 well-covered genomes in the SYNTH181 dataset (with coverage above 5x), while all other methods had coverage below 75% (Supplementary Note 2). Similarly, metaFlye/metaFlye$_{strain}$ produced the most contiguous assemblies of the entire metagenome (NGA20=1.25 Mbp/1.23 Mbp), followed by Canu (923 kbp), miniasm (782 kbp), Flye (347 kbp), and wtdbg2 (341 kbp). Similar conclusions were made from analyzing the smaller SYNTH64 community, with metaFlye producing assemblies with better reference coverage and NGA50 (Extended Data Figure 1, Supplementary Note 2). Flye (in single genome mode), produced inferior assemblies on both synthetic datasets.

Analyzing HMP assemblies.

The Human Microbiome Project (HMP) mock dataset represents a mock human gut microbiome formed by 22 bacteria with known reference genomes sequenced using PacBio reads (total length 6.8 Gbp and N50 = 6.7 kbp). Nineteen of these bacteria have read coverages ranging from 39x (*B. cereus*) to 477x (*H. pylori*). Since the remaining three genomes (*M. smithii, C. albicans,* and *S. pneumoniae*) have low coverage (below 1x), they were excluded from further analysis.

We used metaQUAST [33] to evaluate the statistics of the combined references (Table 1; Supplementary Table 3, Extended Data Figure 2, Supplementary Note 3) as well as to compute the separate statistics for each species present in the sample (Figure 3; Extended Data Figure 3). Because miniasm outputs contigs with a high per-nucleotide error rate, we performed one round of contig polishing using Racon [34].

The metaFlye, Canu, and miniasm assemblies had the highest NGA50 (2.0 Mb, 1.8 Mbp and 1.8 Mbp, respectively) and highest reference coverage (>99.6%). The wtdbg2 and FALCON assemblies had reduced reference coverage and lower contiguity, associated with bacteria with abundances substantially deviating from the median dataset coverage (*B. cereus, R. shaeroides, C. beijerinckii* and *H. pylori;* Figure 3). Miniasm and metaFlye contigs had the fewest number of misassemblies (71 and 72, respectively), followed by wtdbg2 (105), Canu (105) and FALCON (116). metaFlye assembled all 14 known plasmids that have been previously identified in the HMP dataset. In comparison, metaplasmidSPAdes short-read plasmid assembler failed to assemble seven out of the 14 plasmids from the same sample [35]. Miniasm, Canu, FALCON, and wtdbg2 failed to assemble one, two, four, and four plasmids, respectively. As expected, Flye (in single genome mode) produced less contiguous assembly (NGA50=1.4 Mbp) and had more misassemblies (100), as compared to metaFlye.

Analyzing Zymo assemblies.

The ZymoBIOMICS Microbial Community Standards datasets represent mock community datasets generated using ONT reads with an N50 of ~5 kbp [5, 36]. The *ZymoEven* mock community consists of eight bacteria with abundance ~12% and two yeast species with abundance ~2%. The *ZymoLog* dataset represents the same microbial community with abundances distributed as a log scale (Figure 3). Each of the two communities was sequenced using GridION (total read lengths of 14 Gbp and 16 Gbp for the ZymoEven and ZymoLog datasets, respectively) and PromethION (total read lengths of 146 Gbp and 148 Gbp for the ZymoEven and ZymoLog datasets, respectively). Since the provided reference of the *S. cerevisiae* was highly fragmented (N50 = 8 kbp), we substituted them with the closest complete reference strain from NCBI (JEC21). Because of the structural differences between the references and the assembled strains, we ignored misassemblies from *S. cerevisiae* and *C. neoformans* genomes in the total count of the misassemblies (Supplementary Note 3).

The metaFlye and Canu assemblies of the ZymoEven GridION covered 95.7% and 94.9% of the references and improved over the miniasm and wtdbg2 assemblies (80.1% and 75.2%, respectively). The lower coverage of miniasm and wtdbg2 is primarily explained by the reduced performance on two yeast species, as compared to the bacterial genomes (Figure 3). metaFlye, as compared to Canu, had slightly better NGA50 on bacterial genomes (Figure 3), and had fewer total bacterial misassemblies (7 and 11, respectively). Flye, as compared to metaFlye, produced bacterial genomes with similar contiguity, but failed to assemble both yeast genomes (with substantially lower read coverage).

The ZymoLog GridION dataset contains only four species with coverage above 3x: *L. monocytogenes* (3960x), *P. aeruginosa* (158x), *B. subtilis* (38x) and *S. cerevisiae* (7x). metaFlye and Canu reconstructed over 99% of the three bacteria and 79% and 76% of the *S.cerevisiae* genome, respectively. Miniasm and wtdbg2 assembled smaller fractions of *S.cerevisiae* (11% and 40%, respectively). Canu and metaFlye had the best overall contiguity (NGA25=81 kbp and 75 kbp, respectively). Flye failed to produce any assembly of this dataset due to poor *k*-mer indexing (Methods).

metaFlye assembly of ZymoEven PromethION dataset had comparable reference coverage and contiguity to the GridION assembly. In contrast, for the ZymoLog dataset, the reference coverage of metaFlye assembly increased from 46% to 58%, and NGA25 increased from 75 kbp to 3.5 Mbp (Table 1, Figure 3) - a result of the increased read coverage of species with low abundance. wtdbg2 resulted in assemblies with reduced reference coverage and contiguity, as compared to metaFlye (Table 1). Canu and miniasm failed to produce PromethION dataset assemblies due to either runtime or memory requirements (Supplementary Note 3).

Assembly of the sheep gut microbiome.

To investigate the capability of long-read metagenomics to recover complete bacterial genomes from complex samples, we have sequenced a sheep fecal sample using PacBio CCS protocol (Methods). We generated ~3.7 million reads (49.2 Gbp of sequence) with read N50 ~14 kbp after the CCS consensus calling. metaFlye assembly yielded 1.4 Gbp of sequence in contigs longer than 10 kbp (1 Gbp in contigs longer than 100 kbp), including 192 contigs longer than 1 Mbp with total length 344 Mbp (Table 2). 28 of these contigs were circular, likely representing complete bacterial genomes. In addition, there were 59 simple connected components (>1 Mbp in length with fewer than 10 edges) that represent partial or complete bacterial genomes with a relatively small number of repeats.

In comparison, Canu assembled more sequence in short contigs (1.5 Gbp vs 1.4 Gbp in contigs longer than 10 kbp), but less sequence in long contigs (0.9 Gbp vs 1 Gbp in contigs longer than 100 kbp). Wtdbg2 and miniasm produced assemblies with lower contiguity and the total length, as compared to metaFlye and Canu (Supplementary Table 6).

CheckM v1.1.2 [37] analysis of conserved taxonomy markers predicted 63 contigs to be >90% complete and <5% contaminated in the metaFlye assembly, potentially representing complete or nearly-complete bacterial genomes (25 out of these 63 contigs were circular). In comparison, Canu assembled 49 such contigs. Out of contigs that were >90% complete, 8.6% metaFlye contigs and 9.0% Canu contigs were reported to have >5% contamination, suggesting a low chimerism rate of both assemblies. In addition, we investigated the quality of contigs containing multiple 16S rRNA gene copies (Methods). Out of 223 metaFlye contigs with two or more 16S rRNA gene copies, 211 contained at least 97% similar 16S rRNA copies (a level of similarity expected within bacterial species), confirming the low chimerism rate (Supplementary Note 4).

Prodigal [38] predicted slightly more ORFs in the Canu assembly (1,569,745 vs 1,503,966 for metaFlye), however, the clustering of the ORF sequences at 99% similarity revealed slightly more clusters for metaFlye (1,387,782 vs 1,350,688 for Canu). This could be explained by an increased amount of sequence duplication in the Canu assembly. This distribution of ORF lengths and GC content was similar in both assemblies (Extended Data Figure 4). metaFlye assembly contained fewer split-reads, indicating better local sequence quality (Supplementary Table 4). plasmidVerify [35] identified 143 putative plasmids in metaFlye assembly and only 12 plasmids in Canu assembly (Methods). In addition, viralVerify (https://github.com/ablab/viralVerify) identified 284 and 183 putative viruses in the metaFlye and Canu assemblies, respectively.

We performed a taxonomic assignment of each contig with the BlobTools pipeline [39], which uses DIAMOND alignments [40] against the UniProt reference proteomes database [41] (accessed December 2019). Most of the metaFlye contigs were identified as being of Bacterial (1.4 Gbp), Eukaryotic (47 Mbp), and Archaeal (33 Mbp) origins (Extended Data Figure 5, Supplementary Table 5). Interestingly, 23 Mbp out of 47 Mbp of the Eukaryotic-origin contigs were further assigned to the *Nematoda* phylum. This was consistent with the necropsy report of the animal, which revealed the evidence of parasite infection (Methods).

metaFlye detected 1873 simple bubbles, 166 roundabouts, and 95 superbubbles of sizes ranging from 0.5 kbp to 50 kbp in this dataset, including a single bacterial genome of *Clostridia* class with 20 simple bubbles and 10 superbubbles, illustrating its complex strain composition (Figure 4; Methods).

Analyzing human microbiome assemblies.

A recent study [6] introduced a metagenome assembly pipeline OPERA-MS that combines short- and long-read assembly with clustering of metagenome-assembled genomes using the available bacterial references. The authors showed that OPERA-MS improves assembly contiguity by an order of magnitude as compared to short read-only methods. To benchmark the performance of long-read assemblers on these human gut datasets, we extracted all available records from the ENA database (project ID: PRJEB29152) and excluded three samples where Canu failed (two samples) or metaFlye failed (one sample). Removing these samples resulted in 19 datasets (Supplementary Table 9) with total read lengths varying from 1.6 Gbp to 8.0 Gbp.

We used metaFlye, Canu, miniasm, and wtdgb2 to assemble each dataset separately (Supplementary Table 10), followed by polishing with the corresponding Illumina reads using Pilon [42]. metaFlye and Canu assembled 837 and 815 Mbp of sequence in contigs >10 kbp, and 152 and 125 Mbp in contigs >1 Mbp, respectively. Miniasm and wtdbg2 produced suboptimal assemblies that were substantially shorter (377 Mbp and 684 Mbp, respectively), and had fewer 90%-complete contigs (Supplementary Table 7). Table 2 summarizes the reference-free benchmarks of metaFlye and Canu assemblies. In brief, metaFlye has produced more 90%-complete contigs (14), had a higher rate of contigs validated using 16S rRNA (77 out of 100). and recovered more plasmids (109) and viruses (49), as compared to Canu. metaFlye identified 1141 simple bubbles, 78 superbubbles, and 354 roundabouts of sizes ranging from 0.5 kbp to 50 kbp in this dataset (Extended Data Figure 6).

OPERA-MS implements a hybrid approach that initially assembles short-read contigs and then uses long reads to scaffold these contigs. This strategy has resulted in longer, but less contiguous assembly (Supplementary Table 7) with only one 90%-complete contig and only sixteen complete 16S rRNA genes (while metaFlye and Canu reconstructed 852 and 1,091 complete 16S rRNA genes, respectively).

We further used SibeliaZ [43] to analyze the sequence overlap between the samples (Methods), and found that 159 Mbp (~40%) of the total sequence generated by metaFlye for all 19 samples appears in at least two samples (Methods, Extended Data Figure 7). We therefore performed co-assembly by running metaFlye on the mix of reads from all samples (Methods).

Search for novel biosynthetic gene clusters in human gut assemblies.

Non-Ribosomal Peptides (*NRPs*) are biomedically important natural products that include many antibiotics [44]. Most NRPs are *cyclopeptides* synthesized via *non-ribosomal* (rather than genetic) code and built from over 300 different amino acids. Searching for new NRPs is an important goal since many pathogens have developed resistance against most drugs, including daptomycin and vancomycin, NRP antibiotics of last resort [45]. Today, little is known about antibiotic NRPs that are produced by bacteria that live in the human gut (rather than doctor-prescribed) and it is unclear whether the continuous exposure to them leads to the development of antibiotic resistance.

A recent study [46] introduced the biosyntheticSPAdes tool for identifying NRP-synthesizing *Biosynthetic Gene Clusters* (*BGCs*) in short-read *isolate* assemblies, but, at the same time, acknowledged that short-read *metagenome* assemblies are not adequate for identification of these long (average length ~60 kb) and repetitive (made up of multiple highly similar domains) BGCs. Here we show that metaFlye addresses this limitation and assembles many novel NRP-synthesizing BGCs in the human gut (Supplementary Note 5). This analysis is consistent with the recent discovery of a surprisingly large array of still unknown cyclopeptides in the human gut that are synthesized by still unknown BGCs [47]. We benchmarked OPERA-MS, Canu, and metaFlye and demonstrated that metaFlye co-assembly recovered more known NRP-synthesizing BGCs than the other assemblies (including separate sample assemblies by metaFlye; Supplementary Note 5). metaFlye co-assembly was the only method that resolved all repeats in a known NRP-synthesizing BGC that synthesizes a compound colibactin associated with colorectal cancer [48]. Since these repeats represent adenylation domains (that define the colibactin structure), identification of the complete BGC is a prerequisite for follow-up structure elucidation efforts using peptidogenomics approaches [49].

Analyzing cow rumen assemblies.

To further benchmark metaFlye and the other algorithms, we assembled a cow rumen metagenomic dataset sequenced in a recent study [12], that consists of PacBio CLR reads (total length 52.2 Gbp with N50 ~9 kb) and Illumina reads (Supplementary Note 6). The results are summarized in Table 2 and Supplementary Table 8. Briefly, metaFlye produced the most 25%-complete contigs (16), recovered the highest number of 95% 16S rRNA clusters (115), and had the most contigs validated using 16S rRNA (22 out of 25). None of the assemblers produced contigs with more than 90% completion, likely due to the higher complexity of the cow rumen microbiome, as compared to the sheep and human fecal samples [12].

Discussion

Although long-read metagenomics is a promising direction for untangling complex bacterial communities, it faces difficult algorithmic challenges. We developed the long-read metagenomic assembler metaFlye and benchmarked it using simulated, mock, and real microbial communities. metaFlye assemblies of the HMP and Zymo mock communities had similar or better quality, as compared to the Canu assemblies (in terms of the reference fraction and NGA50 metrics). Both metaFlye and Canu showed substantial improvement over miniasm, wtdbg2, and FALCON on most of the mock community datasets. While miniasm produced a good-quality assembly of the HMP dataset (with relatively uniform species abundance), it failed to assemble substantial fractions of low-abundance species in the Zymo datasets. Similarly, wtdbg2 and FALCON did not recover substantial parts of the HMP and Zymo datasets, and had reduced assembly contiguity. metaFlye was at least 10-fold faster than Canu on all metagenomic datasets we analyzed. Only metaFlye and wtdgb2 were able to scale to the 150 Gbp PromethION runs, but the wtdbg2 PromethION assemblies were substantially more fragmented.

Although mock bacterial communities with known reference genomes are convenient for benchmarking, they do not represent the full complexity of environmental metagenomes. We thus simulated two extra communities of 64 and 181 bacteria with realistic abundances distribution and species composition. Our analysis using the simulated datasets showed that long-read assemblers are facing challenges when assembling (i) genomes with low relative abundance and (ii) genomes with closely related strains or species present in a metagenome. metaFlye showed substantial improvement over Canu, miniasm, and wtdbg2 in assembling these synthetic communities. metaFlye in the strain mode produced more accurate assemblies of the closely-related species and strains at the cost of slightly decreased contiguity.

metaFlye assembly of the sheep microbiome resulted in 63 nearly-complete bacterial contigs, highlighting the power of long-read metagenomics to recover the high-quality genomes from complex microbial communities. metaFlye also improved on Canu, miniasm, and wtdbg2 by producing more contigs with a high degree of completion, and capturing more plasmids and viruses. Importantly, metaFlye enables the analysis of bacterial strains through identifying alternative strain structures, while other assemblies do not retain the strain information.

The analysis of human microbiome samples discovered ten NRP-synthesizing BGCs in metaFlye assemblies, including BGC producing Acinetobactin, Colibactin, and Paenibacterin. In contrast, short-read metagenomic assemblies rarely capture any (long and highly repetitive) NRP-synthesizing BGCs, which makes the downstream NRP discovery difficult [22].

Methods

Assembling mock communities and simulated datasets.

metaFlye v2.7-b1589 (commit fbd6ba5) was run using the *"--meta --plasmids"* options for HMP, SYNTH64 and SYNTH181 datasets. We added an option "*--min-overlap 2000*" to assemble Zymo GridION datasets to compensate for shorter read length.

Canu v1.9 was run using parameters recommended for metagenome assembly on the HMP, Zymo and SYNTH datasets: "*corOutCoverage= 10000 corMhapSensitivity=high corMinCoverage=0 redMemory=32 oeaMemory=32 batMemory=200"*. We note that running Canu with default parameters is faster than running it with metagenomic parameters (114 versus 756 CPU hours to assemble the HMP mock dataset). However, the default parameters produce nonoptimal assemblies of species with low abundance: e.g. the assemblies of *B. cereus, C.beijerinckii,* and *R.sphaeroides* in the HMP dataset were substantially more fragmented, as compared to the metagenomic parameters set. According to the documentation, Canu outputs circular contigs with overlapping ends (multiple kbp in size), which were reported as misassemblies by QUAST. To prevent this, we post-processed HMP, Zymo and SYNTH assemblies by trimming the overlapping ends of circular contigs output by Canu.

Miniasm 0.3 was run using its default parameters on the HMP, Zymo and SYNTH datasets, followed by polishing using Racon v1.4.10 [34]. FALCON (pb-falcon 0.2.5) was run using a configuration file recommended for bacterial assemblies. Wtdbg2 v2.3 was run using the default parameters for the HMP dataset. However, since the Zymo datasets had higher read coverage as well as low-abundance species, we increased the *k*-mer frequency coverage range using "*--node-max 1000 -e 2*" as suggested by the developers. This resulted in an increase in the total assembly length as compared to the default settings (from 28 Mbp to 55 Mbp for the ZymoEven dataset, and from 12.6 Mbp to 23.4 Mbp for the ZymoLog dataset). We used the default parameters for the SYNTH datasets, and additionally polished the assemblies using Racon v1.4.10.

All tools were benchmarked on a computational node with two Intel Xeon 8164 CPUs, with 26 cores each and 1.5 TB of RAM.

Generating assemblies of real metagenomic datasets.

We used metaFlye v2.7b (commit a52dfba) with "*--meta --plasmids*" options to generate all real metagenomic assemblies. The "*--min-overlap*" parameter was set to 2 kbp for the cow rumen (otherwise, automatically selected). We found that 13% of PacBio reads in the cow rumen dataset contained more than one PacBio subread (reads with multiple polymerase passes). We split such chimeric reads using the pbclip tool (https://github.com/fenderglass/pbclip) before running metaFlye.

We ran Canu v1.8 on the human gut dataset and Canu v1.9 on the sheep gut microbiome dataset using the metagenomic parameters described above. For the sheep gut microbiome dataset that consists of PacBio CCS reads (estimated error rate ~2%), we used "-pacbio-corr" mode to generate assemblies. In addition, we tested "-pacbio-hifi" mode (recently introduced in Canu v1.9), which resulted into assembly with increased contiguity, but high chimera rate (~20% contigs with >90% completeness had >5% contamination rate as reported by CheckM). We thus selected the assembly produced with "-pacbio-corr" for our analysis.

Miniasm v0.3 and wtdbg2 v2.3 were run using the default parameters on the cow rumen, human gut and sheep microbiome datasets. We applied long-read polishing using Racon v1.4.10 to both miniasm and wtdbg2 assemblies to improve the base quality.

Sequencing of the sheep microbiome.

Sheep from the flock maintained at the U.S. Meat Animal Research Center (USMARC), are monitored for health. Necropsy is performed in some cases if the cause of death is uncertain. Necropsy of one wether in 2018, revealed evidence of infection with coccidial single-cell parasites and strongyloides nematode parasites. Fecal matter was collected from the colon of this animal, with watery texture consistent with diarrhea and the presence of eggs presumed to reflect parasite infection.

DNA was extracted from the fecal material using the QIAamp PowerFecal DNA kit as suggested by the manufacturer (Qiagen), including the bead beating step with a Tissuelyzer. The success of the preparation of high molecular weight DNA was confirmed using Fragment Analyzer (Advanced Analytical Technologies). The DNA was sheared to fragment size in the 9–18 kbp range using Digilab Genomic Solution Hydroshear instrument (Digilab), and sequencing libraries were prepared using the SMRTbell Template Prep Kit v1.0 as recommended (Pacific Biosciences). The libraries were size-selected using the SAGE ELF size selection system (Sage Science) to final target size, which varied from 9 kbp up to 16 kbp. Sequencing was performed on a Sequel instrument (Pacific Biosciences) using v2.1 chemistry (libraries in the 9–10 kbp range) or v3.0 chemistry (libraries in the 12–16 kbp range) and 20-hour movies (6 8-hour pre-extension). A total of 45 SMRT cells were collected using 10 individual library preparations (4 selected at 9–10 kbp; 3 selected at 12–13 kbp; 3 selected at 15–16 kbp). Following sequencing, polymerase reads were converted to circular consensus reads using the CCS application in SMRT Link software v6.0 and default settings. The sequenced sample was fully consumed during the experiment.

Identifying putative plasmids and viruses.

We used plasmidVerify [35], commit 69e2092b and viralVerify (https://github.com/ablab/viralVerify), commit 017d43a2 to identify putative plasmids and viruses. We only considered contigs that were (i) circular and (ii) shorter than 500 kbp as potential plasmid and viral candidates to reduce the number of false positives matches (representing fragmented plasmids and viruses).

Strain statistics for the metaFlye sheep microbiome assembly.

The bacterial genome illustrated in Figure 4a was identified as *Clostridia* class by comparing the extracted 16S rRNA sequences against the SILVA database [50] to identify the closest database match with 84% identity. We ran metaFlye with the “--keep-haplotypes” option, visualized the assembly graph with Bandage [51], and visualized the simple bubble statistics using Matplotlib [52] and Seaborn (https://seaborn.pydata.org/). Sequence identity was estimated from the Jaccard similarities [53]. ORF sequences were clustered at 99% similarity using CD-HIT [54].

Validating assemblies using 16S rRNA genes.

Complete 16S rRNA genes were predicted using Barrnap v0.9 (https://github.com/tseemann/barrnap). We further clustered these genes at 95% identity using vsearch v2.14.1 [55] to reveal the fine-grained taxonomic composition of the microbial communities. Singletons were removed because they can potentially represent poorly polished copies of 16S rRNA genes rather than separate 16S rRNA genes (and artificially inflate the number of discovered clusters). To validate the structural accuracy of contigs, we clustered 16S rRNA copies within each contig at 97% diversity (expected for single bacterial species) using vsearch.

Analyzing human gut sample composition overlap.

We used SibeliaZ [43] v1.2.0 with parameters “-k 25 -n -f 50” to generate multi-way whole-genome alignments between all assembled samples. Each alignment block represents the aligned sequence that appears in one or multiple samples. *Non-redundant* sequence [56] was computed by collapsing each multi-way aligned region into a single consensus. metaFlye and Canu assemblies contained 425 Mbp and 393 Mbp of non-redundant sequence, respectively (Extended Data Figure 7). 159 Mbp (~40%) of the non-redundant metaFlye sequence appears in multiple samples, and 266 Mbp was unique to a single sample.

Co-assembly of multiple human gut samples.

Since there is a large sequence overlap between human gut samples, we co-assembled all of them by running metaFlye on the mix of reads from all samples. Co-assembly is computationally more difficult than assembling each sample separately due to (i) increased strain divergence levels and (ii) increased shared sequence content that complicates the assembly graph. Indeed, the total number of detected simple bubbles, superbubbles, and roundabouts increased from 1573 (separate metaFlye assemblies) to 2873 (co-assembly), revealing richer strain composition. Nevertheless, metaFlye co-assembly resulted in 453 Mbp of sequence, which closely matched the amount of non-redundant sequence from assemblies of separate samples. We also attempted to run Canu on the mix of all reads but terminated the pipeline after no substantial progress within a month of running it on a computational server.

Solid *k*-mer selection in metagenome assemblies.

The Flye algorithm [16] selects solid *k*-mers as follows (the typical *k*-mer size is 15 or 17 nucleotides for PacBio and ONT reads). In the first pass through all reads, the algorithm counts frequencies of *k*-mer hashes using a fixed-size array of counters. In the second pass, *k*-mers with pre-computed frequencies higher than a threshold (typically equal to 2 or 3) are counted using the cuckoo hash table [57]. Given the computed *k*-mer frequency table and an estimated genome size $|G|$, the algorithm selects the $|G|$ most frequent *k*-mers and sets a frequency threshold t as the minimum frequency among the selected *k*-mers. The selected threshold t separates solid *k*-mers (that are indexed) from erroneous ones (that are discarded).

This strategy typically results in a relatively small misclassification rate; e.g., in a typical isolate bacterial project only ~5% of unique *genomic* *k*-mers (true *k*-mers from the genome) are missing from the set of solid *k*-mers, and only ~10% of unique solid *k*-mers represent non-genomic *k*-mers. However, although it works well in genomic assemblies, it is not suitable for metagenomic assemblies, because there is no frequency threshold that robustly separates genomic from non-genomic *k*-mers (due to the uneven species coverage). To address this challenge, some short-read metagenomic assemblers use more sophisticated strategies for selecting *k*-mers, such as the *mercy-kmer* approach in MEGAHIT [23]. However, since these approaches do not work for long reads, we describe an alternative strategy for solid *k*-mer selection and benchmark it using both isolate and metagenome datasets.

Similarly to the uniform coverage mode in Flye, metaFlye also starts with counting *k*-mers in all reads. Although high-frequency *k*-mers are still expected to represent genomic *k*-mers, non-genomic *k*-mers arising from reads in high-abundance species often outnumber genomic *k*-mers from low-abundance species. Given a per-nucleotide error rate ε in reads, we estimate the probability of a *k*-mer in a read to be error-free as $E = e^{-k\varepsilon}$, under a Poisson error distribution model. Thus, the expected number of solid *k*-mers in a read is E

** length(read)*. For each read, metaFlye selects a frequency threshold *f,* so that there are at least *E * length(read) k*-mers in this read with frequency at least *f* and indexes *k*-mers above this threshold using a hash table. Similarly to other *k*-mer counting/indexing tools, metaFlye keeps the canonical representation of each *k*-mer, which is defined as the lexicographical minimum of the forward and reverse-complement of the *k*-mer.

We evaluated the uniform and metagenome k-mer selection modes using an isolate genome dataset and a metagenome dataset, for which true *k*-mers were extracted from the available references. Below we show that for isolate genomes, the metagenome k-mer selection mode in metaFlye only slightly deteriorates as compared to the uniform k-mer selection mode in Flye. However, in the case of metagenomes, the metagenome k-mer selection mode significantly improves upon the uniform k-mer selection mode.

The first set of PacBio reads from an *E. coli* isolate (at 50x coverage) contains 254.2M (million) *k*-mers, out of which 56.7M (22%) are genomic. In the uniform *k*-mer selection mode, Flye indexed 55.3M genomic *k*-mers (97% of all genomic *k*-mers) and 5.0M non-genomic (erroneous) *k*-mers. In the metagenome selection mode, metaFlye indexed 50.3M genomic *k*-mers (89%) and 22M non-genomic *k*-mers.

We further used the HMP mock dataset to evaluate the *k*-mer selection in metagenome mode. We focused on the two least abundant genomes in the mixture - *B. cereus* and *R. sphaeroides* - which had coverage that is 2-fold below the median species coverage. These two bacteria contributed to 83M genomic *k*-mers in the reads. In the uniform coverage mode, Flye selected only 33.2M (40%) of their genomic *k*-mers. In contrast, metaFlye selected 71M (86%) of genomic *k*-mers in the metagenome coverage mode.

The challenge of identifying repeats in metagenome assembly graphs.

In difference from contigs (that are expected to represent contiguous segments of a genome), metaFlye first builds error-prone disjointigs that represent arbitrary paths in the assembly graph but can be generated much faster than traditional contigs. To fix potential misassemblies within disjointigs, Flye constructs the *repeat graph* from disjointigs by collapsing each family of long repeats into a single path in the graph [16]. Each edge of the repeat graph is classified as *unique* (if its sequence appears only once in a single genome) or *repetitive* (if its sequence appears multiple times in a single genome or is shared by multiple genomes). The contiguity of Flye assemblies critically depends on its ability to correctly classify unique and repetitive edges of the assembly graph since this classification is needed for identifying *bridging repeats* [16].

Removing all unique edges from the repeat graph breaks it into connected components that we classify either as *simple* repeats (consisting of a single edge) or *mosaic* repeats consisting of multiple edges [58]. Although Flye correctly identifies the vast majority of simple repeats, classification of edges in mosaic repeats [59] is a more challenging task that remains unsolved in the case of metagenomic assemblies. We note that the problem of repeat detection has been studied for short-read metagenomic graphs [60], but it is unclear how to extend it to long-read analysis.

To improve the classification of repeat edges, Flye uses the *diverged read-paths* approach that analyzes read-paths in the repeat graph (a read-path is a path in the repeat graph that a read traverses). It initially classifies all edges in the repeat graph as unique and checks whether all read-paths through a unique edge continue into a single successor edge (a similar test is done for predecessor edges). If there are multiple successors or predecessors, the edge is re-classified as repetitive.

Although this approach works well in genomic assemblies, it is not suitable for metagenomic assemblies since the edge coverage is not a reliable predictor of the edge multiplicity. Without the coverage test, the read-paths criteria might fail to identify repetitive edges that belong to mosaic repeats, since it checks only immediate predecessors and successors of each edge, e.g., the repetitive edge Y within a mosaic repeat in Figure 1a would be classified as a unique edge. To address this pitfall, we substitute the diverged read-paths approach in Flye by the *iterative repeat detection* approach in metaFlye (described below) to identify repeat edges in the metagenome assembly graph without using the coverage information.

Iterative repeat detection.

Initially, metaFlye classifies all edges in the assembly graph as unique. The algorithm iterates through all edges and re-classifies some edges into repetitive as described below. Thus, at each intermediate iteration, the assembly graph may contain both unique and repetitive edges.

Given a read-path through an edge *e*, metaFlye defines the next *unique* edge in this path as a *successor* of *e* (in contrast to the Flye algorithm that considers *any* edge as a successor). A set of all read-paths through an edge defines a set of successors and we denote a successor edge with maximum *support* as e_{max} (support of an edge is defined as the number of read-paths that traverse this edge). To account for chimeric reads, metaFlye filters out all successors with small support, i.e., each successor edge *e* with $support(e)/support(e_{max}) <$ □ . If a unique edge has multiple successors or predecessors, it is reclassified as repetitive.

The described test is performed iteratively on the entire set of edges until no new edges are reclassified as repetitive. Intuitively, in a mosaic repeat, the first iteration of the test will classify *some* of its edges as repetitive, but consecutive iterations extend the set of repeats (Figure 1a). For a faster convergence of the algorithm, we traverse edges of the graph in the increasing order of their length, as short edges are more likely to be repetitive (two iterations are typically sufficient). The default value □ =0.2 was derived empirically through the evaluations on multiple metagenomic and genomic datasets to minimize the number of classification errors.

We evaluated the repeat detection algorithm using the HMP dataset as follows. We aligned each edge of the repeat graph (before graph simplification) against the combined reference genome using minimap2 [61]. The alignment revealed 79 repetitive and 403 unique edges (repetitive edges have more than one distinct alignment over at least half of the edge length). metaFlye erroneously classified 13 out of 403 (3.2%) unique edges as repetitive, and 2 out of 79 repetitive edges as unique (2.5%). Note that the errors of the first type would not lead to misassembly, but might result in under-assembly. The errors of the second type potentially could lead to misassembly, however, the Flye graph simplification algorithm was designed to be robust against the (rare) repeat misclassifications [16].

Bubbles.

Let $G(V, E)$ be a directed weighted graph with the node-set V and the edge-set E. Given a subset U of its nodes, we define E_U as the edge-set formed by all edges of G that connect nodes in U. We refer to a subgraph with the node-set U and the edge-set E_U as the *U-induced subgraph* of G.

A path in a graph is called *short* if its length does not exceed a threshold *bubbleDiameter* (the default value 50 kb). An edge in a graph is called a *bridge* if its removal increases the number of connected components in the graph. An edge that connects a node in $E \mid U$ to a node in U (a node in U to a node in $E \setminus U$) is called an *entrance (exit) edge* for a U-induced subgraph. An ending node of an entrance edge (a starting node of an exit edge) is called an *entrance* (*exit*) node.

A U-induced subgraph is called a *bubble* is (i) it has a single incoming and a single outgoing edge, (ii) it has no bridges, and (iii) for each edge in this subgraph, there is a short path from the entrance to the exit passing through this edge (compare with the definition of a *blob* in ref. [62]). A bubble is called *simple* if it is formed by two parallel edges and a *superbubble*, otherwise (Figure 1).

Finding simple bubbles.

Simple bubbles, often arising from two strains, are formed by two short parallel edges in the repeat graph (Figure 1b). Since metaFlye collapses edges shorter than the *MAX_SEPARATION* parameter (500 bp by default), some simple bubbles are represented as a pair of loop-edges in the repeat graph. In difference from the concept of a bubble in previous studies [28, 63], metaFlye considers bubbles where the entrance and exit are represented by the same node.

Finding superbubbles.

Many short-read assemblers search for superbubble-like structures, defined empirically through the corresponding algorithmic implementation [62, 64]. Although most assemblers require superbubble subgraphs to be acyclic, a generalization that allows cycles was proposed but has not been implemented in a genome assembler yet [65]. In difference from the previously described assemblers (and in difference from the concept of a superbubble in previous studies [28, 63], metaFlye does not require superbubbles to be acyclic and thus has the ability to analyze repeats inside superbubbles. This is an important distinction since metagenomic superbubbles often contain repeats.

metaFlye considers each edge *startEdge* (and the corresponding node *startNode*) in the repeat graph and attempts to find a bubble that has *startEdge* as its potential entrance. It finds an arbitrary simple path *Path* of length at least *bubbleDiameter* starting at *startNode* and iterates over all intermediate edges in this path. For each intermediate edge *endEdge* *(*and the corresponding *endNode),* metaFlye removes this edge from the graph, launches the Dijkstra algorithm to find shortest paths from *startNode* to all other nodes of the graph, and prematurely terminate it if the distance from *startNode* to the next opened node exceeds *bubbleDiameter.* In the case the algorithm does not terminate prematurely (i.e., the distance from *startNode* to all discovered nodes does not exceed *bubbleDiameter*), we run the "reversed" Dijkstra search starting from *endNode* with the flipped direction of edges and *startEdge* removed. If (i) the reversed Dijkstra search was also successful and (ii) both searches have discovered the same set of nodes and edges, we classify the subgraph discovered by the algorithm as a superbubble with the entrance *startNode* and the exit *endNode.* Although the search for an arbitrary path of length at least *bubbleDiameter* (and follow-up launch of the Dijkstra algorithm) can be time-consuming in theory, in practice this algorithm takes minutes to process large metagenomic datasets, such as the cow rumen dataset with over 1 Gbp of assembled sequence and the repeat graph having over 150,000 edges.

Finding roundabouts.

Alternative strains might share repeated sequences with the other genomes within a metagenome, resulting in *roundabouts* (Figure 1d) that popular short-read metagenomic assemblers, such as metaSPAdes [24] and MEGAHIT [23] do not attempt to simplify. metaFlye identifies and simplifies roundabouts by analyzing read-paths in the repeat graph (read-paths represented by a single read are removed to exclude potentially chimeric reads).

To identify roundabouts, metaFlye iterates through all edges of the repeat graph. For each edge *startEdge,* it analyzes all read-paths through *startEdge* in the graph, considers suffixes of these paths that start at *startEdge,* and selects maximal suffixes (i.e., suffixes that are not contained within other suffixes). If there exists an edge *endEdge* traversed by each maximal suffix, metaFlye trims each maximal suffix by removing all its edges, starting from *endEdge.* Finally, metaFlye identifies a roundabout as a subgraph formed by edges in all shortened maximum suffixes. Note that while roundabouts may represent more complex strain variations than superbubbles, the size of the roundabouts is limited by the read lengths, whereas the superbubbles are identified based on the structure of the repeat graph and irrespectively of reads.

Processing strain groups.

metaFlye identifies *strain groups* (bubbles, superbubbles, and roundabouts) and retains each group in the graph during the following graph simplification steps (such as tip clipping and repeat resolution). It has two strain analysis modes: the standard metaFlye *strain-suppression* mode (each strain group is collapsed into a single edge connecting the entrance and exit nodes of the group before the final contigs are generated) and the metaFlye$_{\text{strain}}$ *strain* mode (retaining the alternative strain structures in the graph) which produces less contiguous assemblies that however are better suited for strain analysis.

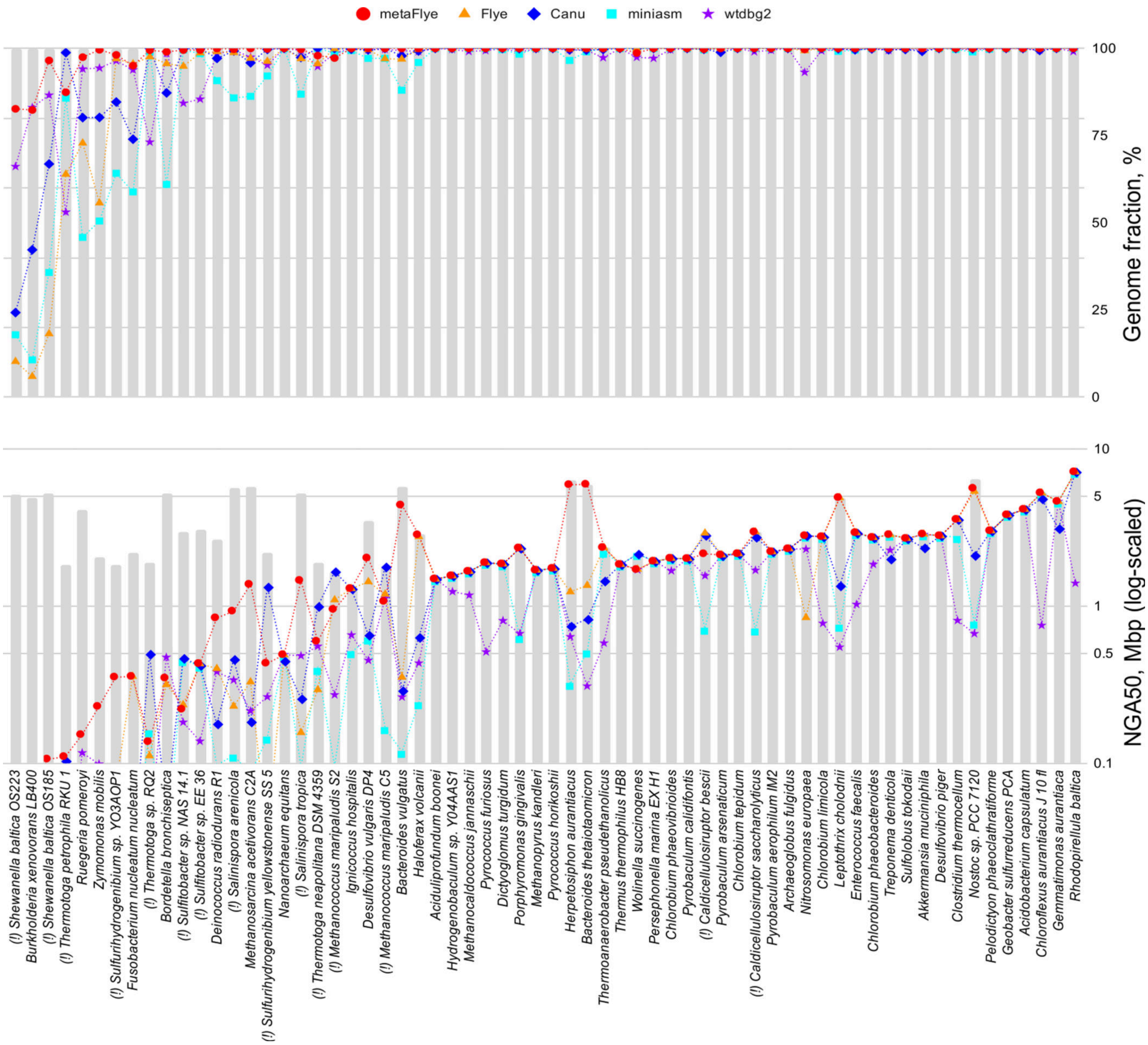
Additional repeat graph simplification procedures.

Some strain variations, such as inversions, do not fall under the definition of bubbles/ roundabouts or are too complex to detect with the described algorithms. After identifying strain groups, metaFlye additionally simplifies the repeat graph by removing edges with locally reduced coverage and long tip edges (Supplementary Note 7).

Assembling short plasmids.

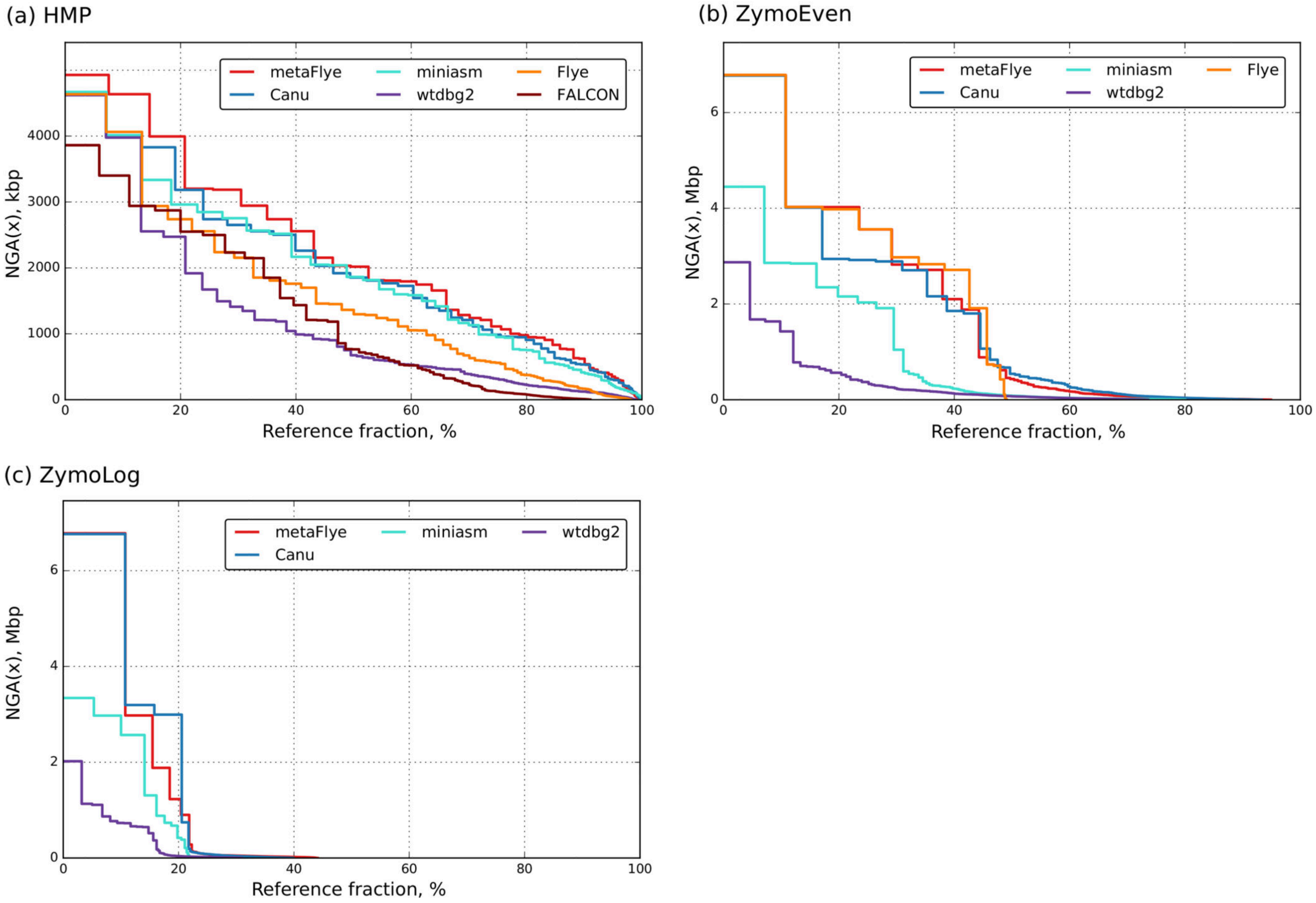
Short plasmid sequencing is an important task since these plasmids represent a large fraction (~30%) of all plasmids in the RefSeq database. However, although existing long-read assemblers perform well in assembling long circular plasmids (longer than the typical read length), our benchmarking revealed that they often miss short plasmids. metaFlye implements an additional module that ensures the assembly of short circular sequences that are spanned by one or two overlapping reads (Supplementary Note 8).

Extended Data



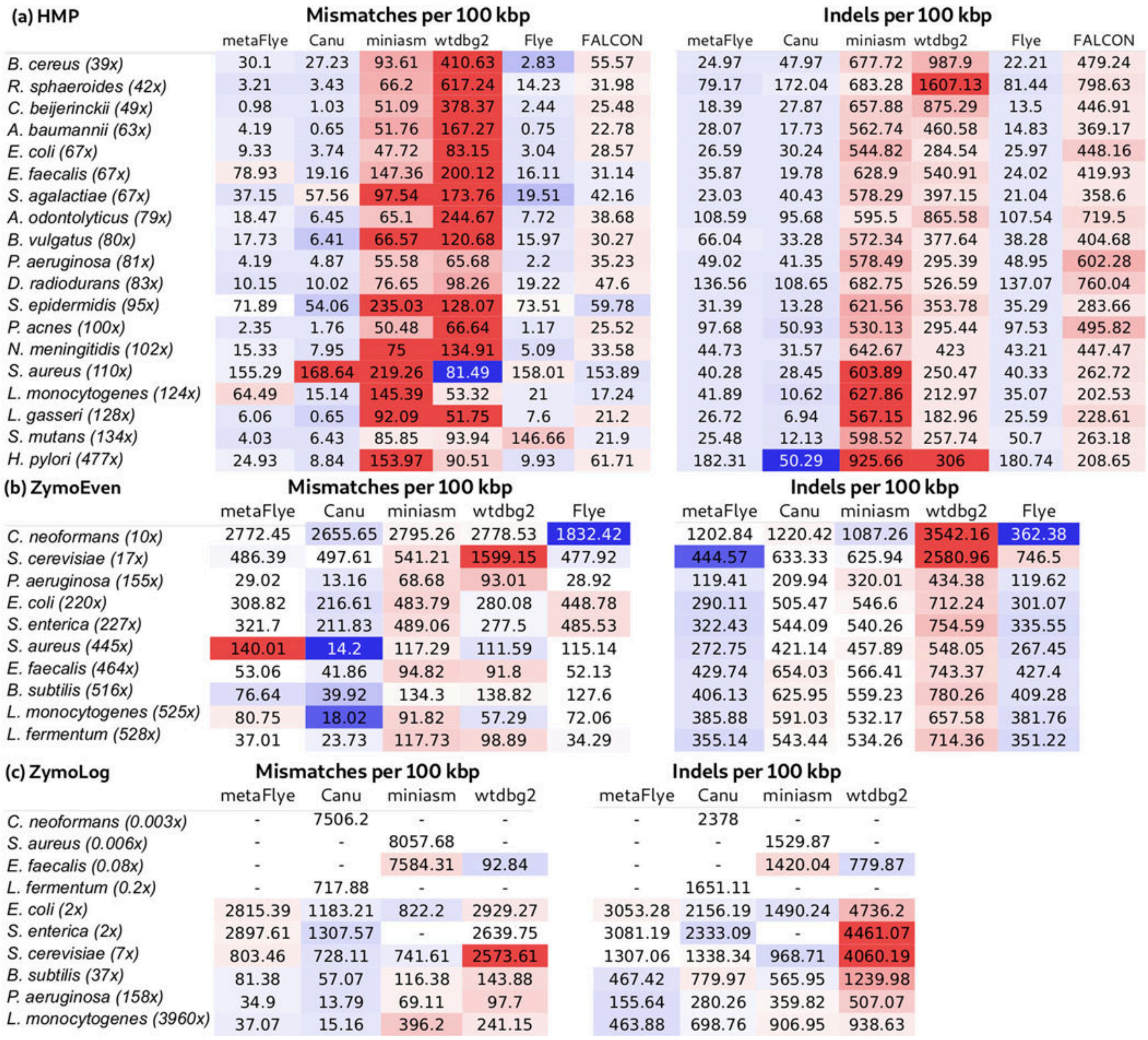
Extended Data Fig. 1. Information about metaFlye, Flye, Canu, miniasm, and wtdbg2 assemblies of the individual genomes in the SYNTH64 dataset.

NGA50 (in megabases) and reference coverage (in percentages) reported for all genomes from the SYNTH64 dataset. Genomes are ordered in the increasing mean NGA50 across all assemblers. Challenging genomes that have closely related species or strains in the metagenome are marked with (!). Grey bars on the NGA50 plot represent the length of the longest chromosome in the reference sequence for each genome (a theoretical upper bound for NGA50). NGA50 is shown in logarithmic scale (not shown for values lower than 100 kb or if the reference coverage is below 50%). The full metaQUAST report for the SYNTH64 dataset is provided in Supplementary Table 1.



Extended Data Fig. 2. NGAx plots for the mock community datasets (HMP mock, ZymoEven GridION, ZymoLog GridION).

NGA(x) is the statistic computed for contigs that are broken at their misassembly breakpoints (if any). NGA(x) is the highest possible number *L* such that all broken contigs that are longer than *L* cover at least *X*% of the reference. Plots were generated by metaQUAST using all available references for each dataset. Flye failed to assemble the ZymoLog datasets due to poor *k*-mer indexing (Methods).



(a) HMP

Mismatches per 100 kbp

	metaFlye	Canu	miniasm	wtdbg2	Flye	FALCON
B. cereus (39x)	30.1	27.23	93.61	410.63	2.83	55.57
R. sphaeroides (42x)	3.21	3.43	66.2	617.24	14.23	31.98
C. beijerinckii (49x)	0.98	1.03	51.09	378.37	2.44	25.48
A. baumannii (63x)	4.19	0.65	51.76	167.27	0.75	22.78
E. coli (67x)	9.33	3.74	47.72	83.15	3.04	28.57
E. faecalis (67x)	78.93	19.16	147.36	200.12	16.11	31.14
S. agalactiae (67x)	37.15	57.56	97.54	173.76	19.51	42.16
A. odontolyticus (79x)	18.47	6.45	65.1	244.67	7.72	38.68
B. vulgatus (80x)	17.73	6.41	66.57	120.68	15.97	30.27
P. aeruginosa (81x)	4.19	4.87	55.58	65.68	2.2	35.23
D. radiodurans (83x)	10.15	10.02	76.65	98.26	19.22	47.6
S. epidermidis (95x)	71.89	54.06	235.03	128.07	73.51	59.78
P. acnes (100x)	2.35	1.76	50.48	66.64	1.17	25.52
N. meningitidis (102x)	15.33	7.95	75	134.91	5.09	33.58
S. aureus (110x)	155.29	168.64	219.26	81.49	158.01	153.89
L. monocytogenes (124x)	64.49	15.14	145.39	53.32	21	17.24
L. gasseri (128x)	6.06	0.65	92.09	51.75	7.6	21.2
S. mutans (134x)	4.03	6.43	85.85	93.94	146.66	21.9
H. pylori (477x)	24.93	8.84	153.97	90.51	9.93	61.71

Indels per 100 kbp

	metaFlye	Canu	miniasm	wtdbg2	Flye	FALCON
B. cereus (39x)	24.97	47.97	677.72	987.9	22.21	479.24
R. sphaeroides (42x)	79.17	172.04	683.28	1607.13	81.44	798.63
C. beijerinckii (49x)	18.39	27.87	657.88	875.29	13.5	446.91
A. baumannii (63x)	28.07	17.73	562.74	460.58	14.83	369.17
E. coli (67x)	26.59	30.24	544.82	284.54	25.97	448.16
E. faecalis (67x)	35.87	19.78	628.9	540.91	24.02	419.93
S. agalactiae (67x)	23.03	40.43	578.29	397.15	21.04	358.6
A. odontolyticus (79x)	108.59	95.68	595.5	865.58	107.54	719.5
B. vulgatus (80x)	66.04	33.28	572.34	377.64	38.28	404.68
P. aeruginosa (81x)	49.02	41.35	578.49	295.39	48.95	602.28
D. radiodurans (83x)	136.56	108.65	682.75	526.59	137.07	760.04
S. epidermidis (95x)	31.39	13.28	621.56	353.78	35.29	283.66
P. acnes (100x)	97.68	50.93	530.13	295.44	97.53	495.82
N. meningitidis (102x)	44.73	31.57	642.67	423	43.21	447.47
S. aureus (110x)	40.28	28.45	603.89	250.47	40.33	262.72
L. monocytogenes (124x)	41.89	10.62	627.86	212.97	35.07	202.53
L. gasseri (128x)	26.72	6.94	567.15	182.96	25.59	228.61
S. mutans (134x)	25.48	12.13	598.52	257.74	50.7	263.18
H. pylori (477x)	182.31	50.29	925.66	306	180.74	208.65

(b) ZymoEven

Mismatches per 100 kbp

	metaFlye	Canu	miniasm	wtdbg2	Flye
C. neoformans (10x)	2772.45	2655.65	2795.26	2778.53	1832.42
S. cerevisiae (17x)	486.39	497.61	541.21	1599.15	477.92
P. aeruginosa (155x)	29.02	13.16	68.68	93.01	28.92
E. coli (220x)	308.82	216.61	483.79	280.08	448.78
S. enterica (227x)	321.7	211.83	489.06	277.5	485.53
S. aureus (445x)	140.01	14.2	117.29	111.59	115.14
E. faecalis (464x)	53.06	41.86	94.82	91.8	52.13
B. subtilis (516x)	76.64	39.92	134.3	138.82	127.6
L. monocytogenes (525x)	80.75	18.02	91.82	57.29	72.06
L. fermentum (528x)	37.01	23.73	117.73	98.89	34.29

Indels per 100 kbp

	metaFlye	Canu	miniasm	wtdbg2	Flye
C. neoformans (10x)	1202.84	1220.42	1087.26	3542.16	362.38
S. cerevisiae (17x)	444.57	633.33	625.94	2580.96	746.5
P. aeruginosa (155x)	119.41	209.94	320.01	434.38	119.62
E. coli (220x)	290.11	505.47	546.6	712.24	301.07
S. enterica (227x)	322.43	544.09	540.26	754.59	335.55
S. aureus (445x)	272.75	421.14	457.89	548.05	267.45
E. faecalis (464x)	429.74	654.03	566.41	743.37	427.4
B. subtilis (516x)	406.13	625.95	559.23	780.26	409.28
L. monocytogenes (525x)	385.88	591.03	532.17	657.58	381.76
L. fermentum (528x)	355.14	543.44	534.26	714.36	351.22

(c) ZymoLog

Mismatches per 100 kbp

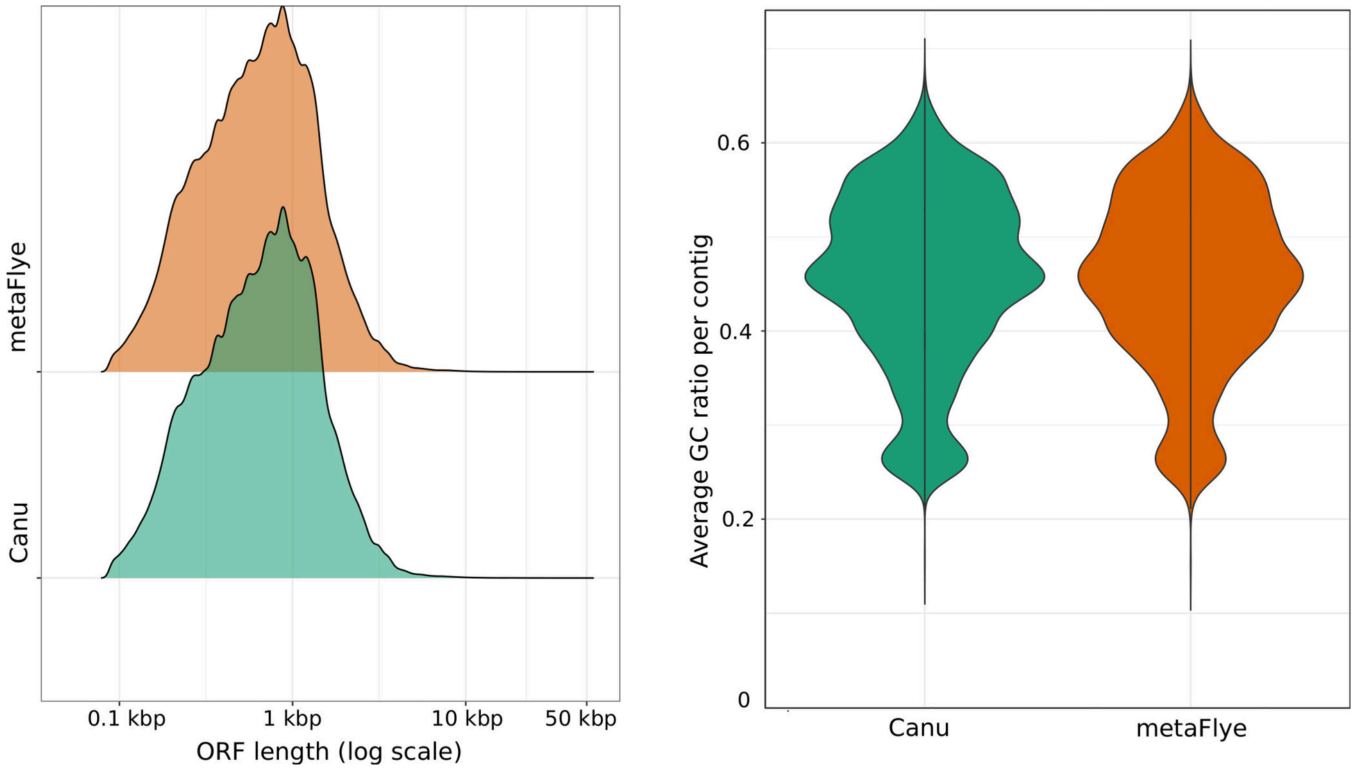
	metaFlye	Canu	miniasm	wtdbg2
C. neoformans (0.003x)	-	7506.2	-	-
S. aureus (0.006x)	-	-	8057.68	-
E. faecalis (0.08x)	-	-	7584.31	92.84
L. fermentum (0.2x)	-	717.88	-	-
E. coli (2x)	2815.39	1183.21	822.2	2929.27
S. enterica (2x)	2897.61	1307.57	-	2639.75
S. cerevisiae (7x)	803.46	728.11	741.61	2573.61
B. subtilis (37x)	81.38	57.07	116.38	143.88
P. aeruginosa (158x)	34.9	13.79	69.11	97.7
L. monocytogenes (3960x)	37.07	15.16	396.2	241.15

Indels per 100 kbp

	metaFlye	Canu	miniasm	wtdbg2
C. neoformans (0.003x)	-	2378	-	-
S. aureus (0.006x)	-	-	1529.87	-
E. faecalis (0.08x)	-	-	1420.04	779.87
L. fermentum (0.2x)	-	1651.11	-	-
E. coli (2x)	3053.28	2156.19	1490.24	4736.2
S. enterica (2x)	3081.19	2333.09	-	4461.07
S. cerevisiae (7x)	1307.06	1338.34	968.71	4060.19
B. subtilis (37x)	467.42	779.97	565.95	1239.98
P. aeruginosa (158x)	155.64	280.26	359.82	507.07
L. monocytogenes (3960x)	463.88	698.76	906.95	938.63

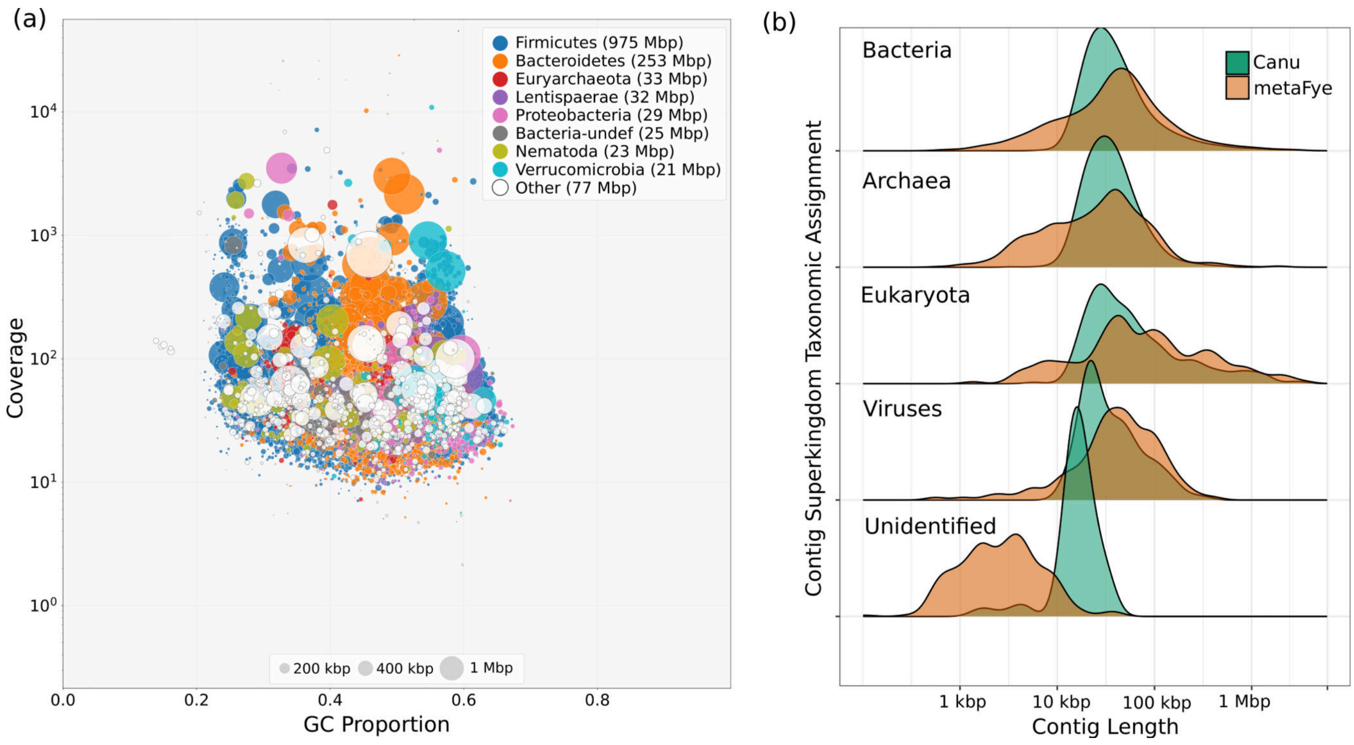
Extended Data Fig. 3. Base-pair accuracy analysis for assemblies of the mock community datasets (HMP, ZymoEven GridION, and ZymoLog GridION).

Heatmaps showing the number of mismatches and short indels per 100 kbp for each species reference, computed using metaQUAST. Blue and red colors correspond to the values higher and lower than the median, respectively. Statistics were not computed for genomes with no assembled sequence ("-" symbol). Flye failed to assemble the ZymoLog datasets due to poor *k*-mer indexing (Methods).



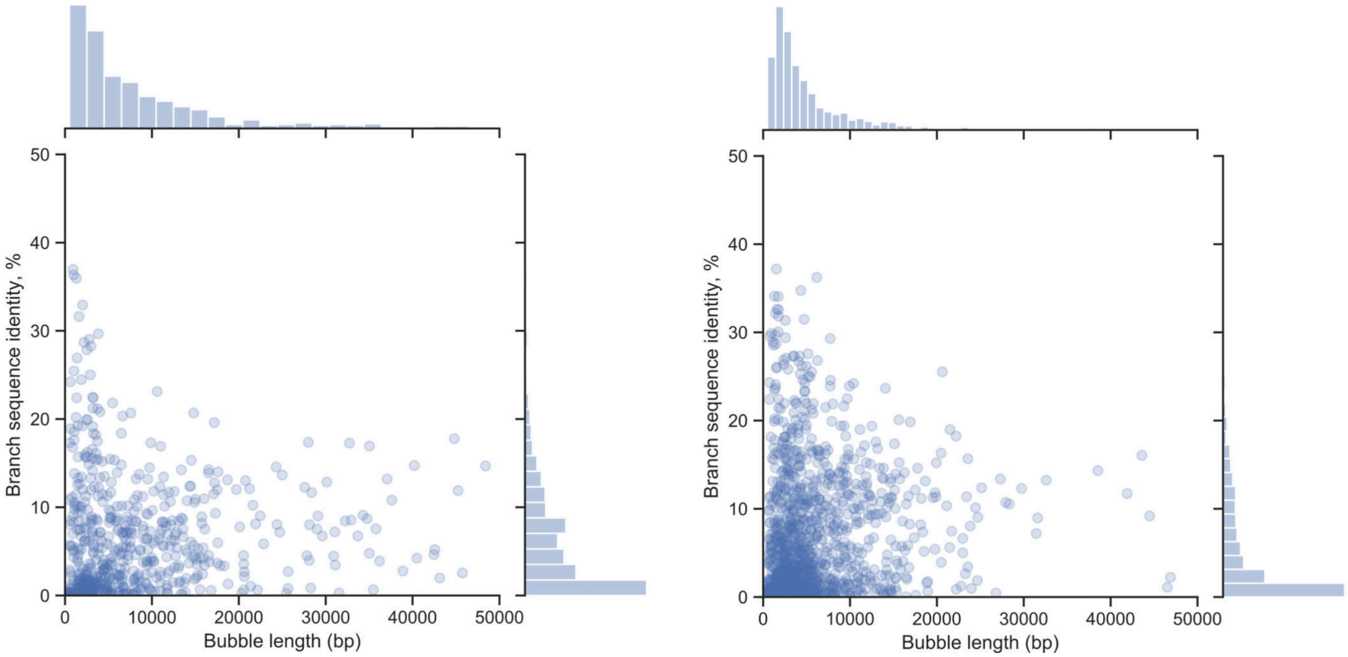
Extended Data Fig. 4. The ORF lengths distribution and the GC content distribution of metaFlye and Canu assemblies of the sheep microbiome.

The ORF length distribution suggests similar base-level accuracy for both assemblies.



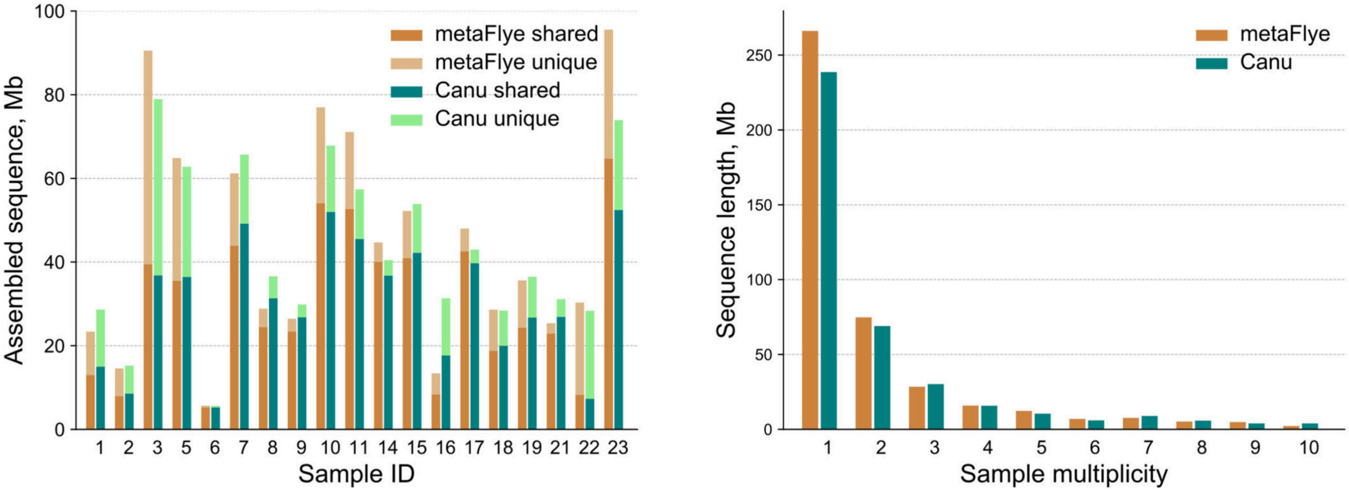
Extended Data Fig. 5. Taxonomic assignments of sheep microbiome assemblies.

(a) metaFlye contigs assignment at the phylum level visualized with BlobTools. (b) Length distributions of metaFlye and Canu contigs within each assigned superkingdom.



Extended Data Fig. 6. Statistics of simple bubbles for the metaFlye assemblies human gut and cow rumen.

(Left) the human gut dataset with 615 bubbles, and (right) the cow rumen dataset with 1510 bubbles. Bubble counts exclude loops, and include roundabouts with two edges.



Extended Data Fig. 7. Analysis of sequence overlap between 19 human gut samples.

Multi-way sequence alignments were computed using SiebliaZ. (left) The proportions of unique and shared sequences in each sample. An assembled segment within a sample is called unique if it has no alignments against sequence from any other samples. Otherwise, the segment is shared. (right) The total amount of sequence for each multiplicity bin. A sequence fragment belongs to the multiplicity bin X if it is shared by exactly X samples.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments.

We are grateful to Denis Bertrand and Niranjan Nagarajan for sharing the metagenomic datasets prior to journal publication. M.K. and P.A.P. were supported by the NSF/MCB-BSF grant 1715911. B.B. was supported by the U.S. National Institutes of Health grant 2-P41-GM103484. D.B. was funded by USDA CRIS project 5090–31000-026–00-D, and K.K., S.S., and T.S. by project 3040–31000-100–00D. A.G. and M.R. were supported by the Russian Science Foundation (grant 19–16-00049). Computational resources were provided in part by the Research Park Computer Center at the St. Petersburg State University.

Data availability.

Sequencing data for the sheep gut sample is available under the NCBI BioProject PRJNA595610. HMP mock dataset is available at: https://github.com/PacificBiosciences/DevNet/wiki/Human_Microbiome_Project_MockB_Shotgun. Zymo datasets: https://github.com/LomanLab/mockcommunity. Cow rumen dataset: NCBI SRA repository under BioProject PRJNA507739. Human stool samples: ENA project PRJEB29152. NCBI accession codes for the sequences used in the NRPS analysis: AM229678.1, AB101202.1, FP929054.1, FP929054.1. All assemblies that were evaluated in this study, as well as SYNTH64 and SYNTH181 datasets are available at: https://doi.org/10.5281/zenodo.3986210 (ref. [66]).

Main Text References

[1]. Jain M, Koren S, Miga KH, Quick J, Rand AC, Sasani TA, Tyson JR, Beggs AD, Dilthey AT, Fiddes IT and Malla S, (2018). Nanopore sequencing and assembly of a human genome with ultra-long reads. Nature Biotechnology, 36(4), p.338.

[2]. Miga Karen H., Koren Sergey, Rhie Arang, Vollger Mitchell R., Gershman Ariel, Bzikadze Andrey, Brooks Shelise et al. (2020) “Telomere-to-telomere assembly of a complete human X chromosome.” Nature 10.1038/s41586-020-2547-7

[3]. Tsai YC, Conlan S, Deming C, Segre JA, Kong HH, Korlach J, ... & NISC Comparative Sequencing Program. (2016). Resolving the complexity of human skin metagenomes using single-molecule sequencing. MBio, 7(1), e01948–15.

[4]. Driscoll CB, Otten TG, Brown NM, & Dreher TW (2017). Towards long-read metagenomics: complete assembly of three novel genomes from bacteria dependent on a diazotrophic cyanobacterium in a freshwater lake co-culture. Standards in Genomic Sciences, 12(1), 9. [PubMed: 28127419]

[5]. Nicholls SM, Quick JC, Tang S, & Loman NJ (2019). Ultra-deep, long-read nanopore sequencing of mock microbial community standards. GigaScience, 8(5), 1–9

[6]. Bertrand D, Shaw J, Kalathiyappan M, Ng AHQ, Kumar MS, Li C, Dvornicic M, Soldo JP, Koh JY, Tong C and Ng OT, (2019). Hybrid metagenomic assembly enables high-resolution analysis of resistance determinants and mobile elements in human microbiomes. Nature Biotechnology, 37(8), 937–944.

[7]. Somerville V, Lutz S, Schmid M, Frei D, Moser A, Irmler S, ... & Ahrens CH (2019). Long read-based de novo assembly of low complex metagenome samples results in finished genomes

and reveals insights into strain diversity and an active phage system. BMC Microbiology, 19(1): 143 [PubMed: 31238873]

[8]. Moss EL, Maghini DG, Bhatt AS (2020). Complete, closed bacterial genomes from microbiomes using nanopore sequencing. Nature Biotechnology. 38, 701–707

[9]. Stewart RD, Auffret MD, Warr A, Walker AW, Roehe R and Watson M, (2019). Compendium of 4,941 rumen metagenome-assembled genomes for rumen microbiome biology and enzyme discovery. Nature Biotechnology, 37(8), 953–961.

[10]. Arumugam K, Bagci C, Bessarab I, Beier S, Buchfink B, Gorska A, Qiu G, Huson DH and Williams RB, (2019). Annotated bacterial chromosomes from frame-shift-corrected long read metagenomic data. Microbiome, 7(61).

[11]. Hiraoka S, Okazaki Y, Anda M, Toyoda A, Nakano S, Iwasaki W (2019) Metaepigenomic analysis reveals the unexplored diversity of DNA methylation in an environmental prokaryotic community. Nature Communications, 10, 159

[12]. Bickhart DM, Watson M, Koren S, Panke-Buisse K, Cersosimo LM, Press MO, Van Tassell CP, Van Kessel JAS, Haley BJ, Kim SW and Heiner C, (2019). Assignment of virus and antimicrobial resistance genes to microbial hosts in a complex microbial community by combined long-read assembly and proximity ligation. Genome Biology, 20(1), 1–18. [PubMed: 30606230]

[13]. Chin CS, Peluso P, Sedlazeck FJ, Nattestad M, Concepcion GT, Clum A, Dunn C, O'Malley R, Figueroa-Balderas R, Morales-Cruz A and Cramer GR, (2016). Phased diploid genome assembly with single-molecule real-time sequencing. Nature Methods, 13(12), 1050–1054 [PubMed: 27749838]

[14]. Li H (2016). Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics, 32(14), 2103–2110. [PubMed: 27153593]

[15]. Koren S, Walenz BP, Berlin K, Miller JR, Bergman NH, & Phillippy AM (2017). Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Research, 27(5), 722–736. [PubMed: 28298431]

[16]. Kolmogorov M, Yuan J, Lin Y, Pevzner PA (2019) Assembly of long, error-prone reads using repeat graphs. Nature Biotechnology 37(5) 540–546

[17]. Ruan J and Li H, (2020). Fast and accurate long-read assembly with wtdbg2. Nature Methods. 17, 155–158 [PubMed: 31819265]

[18]. Goltsman DSA, Sun CL, Proctor DM, DiGiulio DB, Robaczewska A, Thomas BC, Shaw GM, Stevenson DK, Holmes SP, Banfield JF, Relman DA (2018) Metagenomic analysis with strain-level resolution reveals fine-scale variation in the human pregnancy microbiome. Genome Research 28:1467–1480 [PubMed: 30232199]

[19]. Guo J, Wang Q, Wang X, Wang F, Yao J, Zhu H Horizontal gene transfer in an acid mine drainage microbial community (2015) BMC Genomics 16:496 [PubMed: 26141154]

[20]. Eloe-Fadrosh EA, Paez-Espino D, Jarett J, Dunfield PF, Hedlund BP, Dekas AE, Grasby SE, Brady AL, Dong H, Briggs BR, Li WJ, Goudeau D, Malmstrom R, Pati A, Pett-Ridge J, Rubin EM, Woyke T, Kyrpides NC, Ivanova NN (2016) Global metagenomic survey reveals a new bacterial candidate phylum in geothermal springs. Nature Communications, 7, 10476

[21]. Suzuki Y, Nishijima S, Furuta Y, Yoshimura J, Suda W, Oshima K, Hattori M and Morishita S, (2019) Long-read metagenomic exploration of extrachromosomal mobile genetic elements in the human gut. Microbiome, 7(1), 119. [PubMed: 31455406]

[22]. Stevenson LJ, Owen JG, Ackerley DF (2019) Metagenome Driven Discovery of Nonribosomal Peptides. ACS Chem. Biol. 14, 10, 2115–2126 [PubMed: 31508935]

[23]. 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(10), 1674–1676. [PubMed: 25609793]

[24]. Nurk S, Meleshko D, Korobeynikov A, & Pevzner PA (2017). metaSPAdes: a new versatile metagenomic assembler. Genome Research, 27(5), 824–834. [PubMed: 28298430]

[25]. Truong DT, Tett A, Pasolli E, Huttenhower C, Segata N, (2017). Microbial strain-level population structure and genetic diversity from metagenomes. Genome Research, 27(4), pp.626–638. [PubMed: 28167665]

[26]. Ghurye J, Treangen T, Fedarko M, Hervey WJ, Pop M (2019) MetaCarvel: linking assembly graph motifs to biological variants. Genome Biology 20: 174. [PubMed: 31451112]

[27]. Nijkamp JF, Pop M, Reinders MJT, de Ridder D (2013) Exploring variation-aware contig graphs for (comparative) metagenomics using MaryGold. Bioinformatics 29, 2826–2834. [PubMed: 24058058]

[28]. Onodera T, Sadakane K, Shibuya T (2013) Detecting superbubbles in assembly graphs. In International Workshop on Algorithms in Bioinformatics, pp. 338–348. Springer, Berlin, Heidelberg

[29]. Garg S, Aach J, Li H, Sebenius I, Durbin R, Church G (2019) A haplotype-aware *de novo* assembly of related individuals using pedigree sequence graph. Bioinformatics, btz942.

[30]. Sczyrba A, Hofmann P, Belmann P, Koslicki D, Janssen S, Dröge J, Gregor I, Majda S, Fiedler J, Dahms E and Bremges A, (2017). Critical assessment of metagenome interpretation - a benchmark of metagenomics software. Nature Methods, 14(11), 1063–1071. [PubMed: 28967888]

[31]. Jain C, Rodriguez-R LM, Phillippy AM, Konstantinidis KT and Aluru S, (2018). High throughput ANI analysis of 90K prokaryotic genomes reveals clear species boundaries. Nature Communications, 9(1), 1–8.

[32]. Wick R, (2019). Badread: simulation of error-prone long reads. Journal of Open Source Software, 4(36), 1316.

[33]. Mikheenko A, Prjibelski A, Saveliev V, Antipov D, & Gurevich A (2018). Versatile genome assembly evaluation with QUAST-LG. Bioinformatics, 34(13), i142–i150. [PubMed: 29949969]

[34]. Vaser R, Sović I, Nagarajan N, & Šikić M (2017). Fast and accurate de novo genome assembly from long uncorrected reads. Genome Research, 27(5), 737–746. [PubMed: 28100585]

[35]. Antipov D, Raiko M, Lapidus A and Pevzner PA, (2019). Plasmid detection and assembly in genomic and metagenomic data sets. Genome Research, 29(6), 961–968. [PubMed: 31048319]

[36]. Latorre-Pérez Adriel, Pascual Villalba-Bermell Javier Pascual, and Vilanova Cristina. (2020) “Assembly methods for nanopore-based metagenomic sequencing: a comparative study.” Scientific Reports 10(1), 1–14. [PubMed: 31913322]

[37]. Parks DH, Imelfort M, Skennerton CT, Hugenholtz P, Tyson GW (2015) CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. Genome Research 25 1043–1055. [PubMed: 25977477]

[38]. Hyatt D, Chen GL, LoCascio PF, Land ML, Larimer FW, & Hauser LJ (2010). Prodigal: prokaryotic gene recognition and translation initiation site identification. BMC Bioinformatics, 11(1), 119. [PubMed: 20211023]

[39]. Laetsch DR, Blaxter ML (2017) BlobTools: Interrogation of genome assemblies. F1000Research, 6, 1287

[40]. Buchfink B, Xie C, Huson DH Fast and sensitive protein alignment using DIAMOND. Nature Methods 12 (2015): 59–60. [PubMed: 25402007]

[41]. Consortium UniProt. UniProt: a hub for protein information. Nucleic Acids Research 43, (2014): D204–D212. [PubMed: 25348405]

[42]. Walker BJ, Abeel T, Shea T, Priest M, Abouelliel A, Sakthikumar S, ... & Earl AM (2014). Pilon: an integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS One, 9(11), e112963.

[43]. Minkin I and Medvedev P, (2020). Scalable multiple whole-genome alignment and locally collinear block construction with SibeliaZ. bioRxiv 548123

[44]. Kersten RD, Yang Y-L, Xu Y, Cimermancic P, Nam S-J, Fenical W, Fischbach MA, Moore BS and Dorrestein PC (2011) A mass spectrometry--guided genome mining approach for natural product peptidogenomics, Nature Chemical Biology. 7(11), 794–802. [PubMed: 21983601]

[45]. Ling LL, Schneider T, Peoples AJ, Spoering AL, Engels I, Conlon BP, Mueller A et al. (2015) A new antibiotic kills pathogens without detectable resistance. Nature 517 (7535) 455–459. [PubMed: 25561178]

[46]. Meleshko D, Mohimani H, Tracanna V, Hajirasouliha I, Medema MH, Korobeynikov A and Pevzner PA (2019) BiosyntheticSPAdes: reconstructing biosynthetic gene clusters from assembly graphs, Genome Research, 29(8), 1352–1362. [PubMed: 31160374]

[47]. Behsaz B, Mohimani H, Gurevich A, Prjibelski A, Mark F, Smarr L, Dorrestein PC, Mylne JS, Pevzner PA (2020) De Novo Peptide Sequencing Reveals Many Cyclopeptides in the Human Gut and Other Environments, Cell Systems, 10(1), 99–108 [PubMed: 31864964]

[48]. Wilson MR, Jiang Y, Villalta PW, Stornetta A, Boudreau PD, Carrá A, Brennan CA, Chun E, Ngo L, Samson LD and Engelward BP, (2019). The human gut bacterial genotoxin colibactin alkylates DNA. Science, 363 (6428), eaar7785

[49]. Mohimani H, Pevzner PA (2016) Dereplication, sequencing and identification of peptidic natural products: from genome mining to peptidogenomics to spectral networks. Natural Product Reports 33 (1), 73–86 [PubMed: 26497201]

[50]. Quast C, Pruesse E, Yilmaz P, Gerken J, Schweer T, Yarza P, Peplies J and Glöckner FO, (2012). The SILVA ribosomal RNA gene database project: improved data processing and web-based tools. Nucleic Acids Research, 41(D1), D590–D596. [PubMed: 23193283]

[51]. Wick RR, Schultz MB, Zobel J and Holt KE, (2015). Bandage: interactive visualization of de novo genome assemblies. Bioinformatics, 31(20), 3350–3352. [PubMed: 26099265]

[52]. Hunter JD Matplotlib: (2007) A 2D graphics environment. Computing in Science & Engineering 9, 90.

[53]. Dolev S, Ghanayim M, Binun B, Frenkel S, Sun YS (2017) Relationship of Jaccard and edit distance in malware clustering and online identification. In 2017 IEEE 16th International Symposium on Network Computing and Applications (NCA), 1–5.

[54]. Fu L, Niu B, Zhu Z, Wu S, Li W (2012) CD-HIT: accelerated for clustering the next-generation sequencing data. Bioinformatics 28, 3150–3152. [PubMed: 23060610]

[55]. Rognes T, Flouri T, Nichols B, Quince C, & Mahé F (2016). VSEARCH: a versatile open source tool for metagenomics. PeerJ, 4, e2584.

[56]. Pruitt KD, Tatusova T, & Maglott DR (2007). NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Research, 35 (suppl.1), D61–D65. [PubMed: 17130148]

[57]. Li X, Andersen DG, Kaminsky M, & Freedman MJ (2014). Algorithmic improvements for fast concurrent cuckoo hashing. In Proceedings of the Ninth European Conference on Computer Systems (p. 27).

[58]. Jiang Z, Tang H, Ventura M, Cardone MF, Marques-Bonet T, She X, Pevzner PA, Eichler EE. (2007). Ancestral reconstruction of segmental duplications reveals punctuated cores of human genome evolution. Nature Genetics, 39(11): 1361–1368. [PubMed: 17922013]

[59]. Bankevich A, Pevzner PA (2020) mosaicFlye: Resolving long mosaic repeats using long error-prone reads. bioRxiv, doi: 10.1101/2020.01.15.908285

[60]. Koren S, Treangen TJ and Pop M (2011). Bambus 2: scaffolding metagenomes. Bioinformatics, 27(21), 2964–2971. [PubMed: 21926123]

[61]. Li H, (2018). Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics, 34(18), 3094–3100. [PubMed: 29750242]

[62]. Nurk S, Bankevich A, Antipov D, Gurevich A, Korobeynikov A, Lapidus A, Prjibelsky A et al. Assembling genomes and mini-metagenomes from highly chimeric reads. J. Comp. Biol. 20 (2013), 714–737

[63]. Brankovic L, Iliopoulos CS Kundu R, Mohamed M, Pissis SP, Vayani F (2016) Linear-time superbubble identification algorithm for genome assembly. Theoretical Computer Science 609: 374–383.

[64]. Zerbino DR, Birney E (2008) Velvet: algorithms for de novo short read assembly using de Bruijn graphs. Genome Research 18 821–829. [PubMed: 18349386]

[65]. Paten B, Eizenga JM, Rosen YM, Novak AM, Garrison E, Hickey G (2018) Superbubbles, ultrabubbles, and cacti. Journal of Computational Biology, 25, 649–663. [PubMed: 29461862]

[66]. Supporting data for the manuscript “metaFlye: scalable long-read metagenome assembly using repeat graphs” (2020, Version 3.0) [Data set]. Zenodo. 10.5281/zenodo.3986210

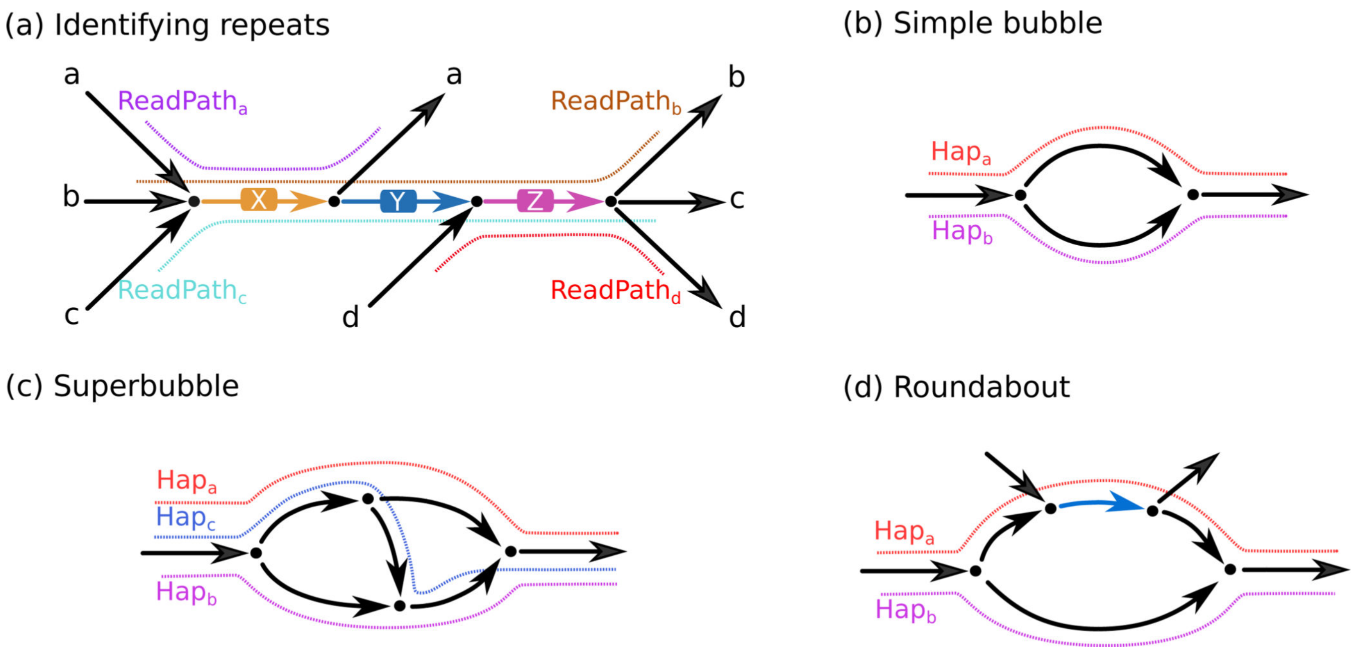


Figure 1. metaFlye repeat annotation and examples of simple bubbles, superbubbles, and roundabouts.

(a) The subgraph of an assembly graph formed by four distinct genome sub-paths. Repeat and unique edges are shown in color and black, respectively. metaFlye identifies edges X, Y, and Z as repetitive by analyzing the distinct read-paths through the sub-graph. (b) A simple bubble formed by two strains. (c) A superbubble formed by three strains. (d) A roundabout formed by two strains, one of which shares a repeat with a different region of the metagenome.

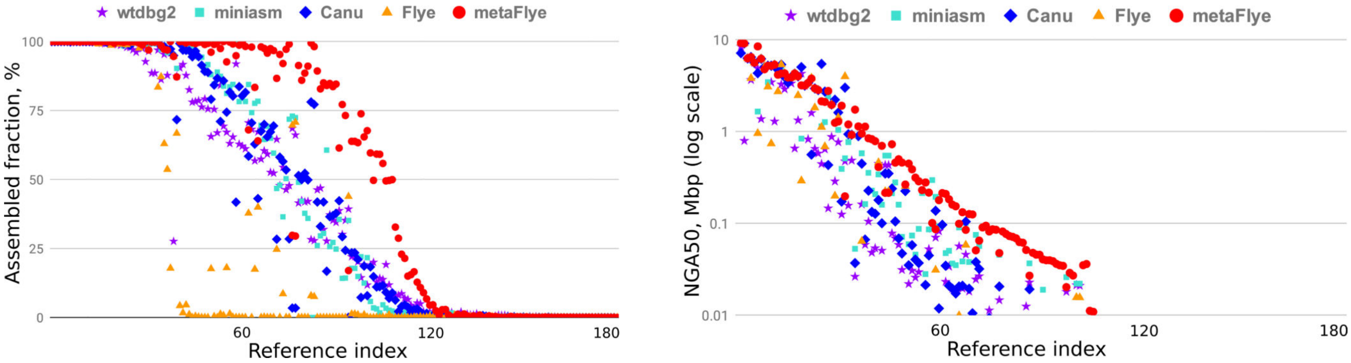
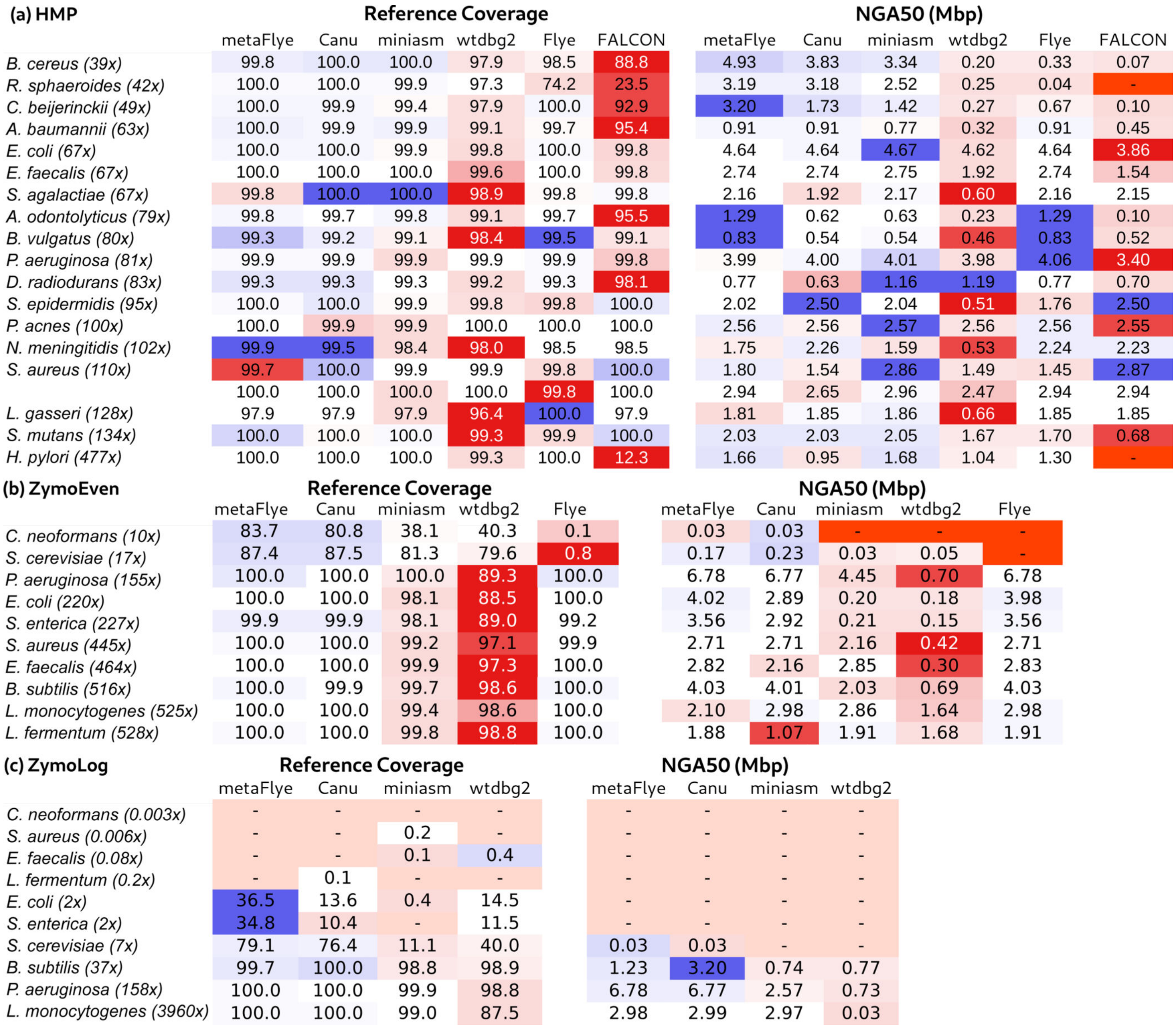


Figure 2. Information about Canu, Flye, metaFlye, miniasm, and wtdbg2 assemblies of the individual genomes in the SYNTH181 dataset.

Assembled fraction and NGA50 are reported for all 181 reference genomes from the simulated dataset. Genomes are ordered in the decreasing mean assembled fraction (left) and NGA50 (right) across five assemblers. NGA50 is the statistic computed for contigs that are broken at their misassembly breakpoints (if any). NGA50 is the highest possible number L such that all broken contigs that are longer than L cover at least 50% of the reference. NGA50 is not shown for values lower than 10 kbp or if the reference coverage is below 50%. 77 (metaFlye), 141 (Flye), 109 (Canu), 106 (miniasm) and 109 (wtdbg2) NGA50 values were filtered this way. The full metaQUAST report is provided in Supplementary Table 2.



(a) HMP

	Reference Coverage						NGA50 (Mbp)					
	metaFlye	Canu	miniasm	wtdbg2	Flye	FALCON	metaFlye	Canu	miniasm	wtdbg2	Flye	FALCON
B. cereus (39x)	99.8	100.0	100.0	97.9	98.5	88.8	4.93	3.83	3.34	0.20	0.33	0.07
R. sphaeroides (42x)	100.0	100.0	99.9	97.3	74.2	23.5	3.19	3.18	2.52	0.25	0.04	-
C. beijerinckii (49x)	100.0	99.9	99.4	97.9	100.0	92.9	3.20	1.73	1.42	0.27	0.67	0.10
A. baumannii (63x)	100.0	99.9	99.9	99.1	99.7	95.4	0.91	0.91	0.77	0.32	0.91	0.45
E. coli (67x)	100.0	100.0	99.9	99.8	100.0	99.8	4.64	4.64	4.67	4.62	4.64	3.86
E. faecalis (67x)	100.0	100.0	100.0	99.6	100.0	99.8	2.74	2.74	2.75	1.92	2.74	1.54
S. agalactiae (67x)	99.8	100.0	100.0	98.9	99.8	99.8	2.16	1.92	2.17	0.60	2.16	2.15
A. odontolyticus (79x)	99.8	99.7	99.8	99.1	99.7	95.5	1.29	0.62	0.63	0.23	1.29	0.10
B. vulgatus (80x)	99.3	99.2	99.1	98.4	99.5	99.1	0.83	0.54	0.54	0.46	0.83	0.52
P. aeruginosa (81x)	99.9	99.9	99.9	99.9	99.9	99.8	3.99	4.00	4.01	3.98	4.06	3.40
D. radiodurans (83x)	99.3	99.3	99.3	99.2	99.3	98.1	0.77	0.63	1.16	1.19	0.77	0.70
S. epidermidis (95x)	100.0	100.0	99.9	99.8	99.8	100.0	2.02	2.50	2.04	0.51	1.76	2.50
P. acnes (100x)	100.0	99.9	99.9	100.0	100.0	100.0	2.56	2.56	2.57	2.56	2.56	2.55
N. meningitidis (102x)	99.9	99.5	98.4	98.0	98.5	98.5	1.75	2.26	1.59	0.53	2.24	2.23
S. aureus (110x)	99.7	100.0	99.9	99.9	99.8	100.0	1.80	1.54	2.86	1.49	1.45	2.87
	100.0	100.0	100.0	100.0	99.8	100.0	2.94	2.65	2.96	2.47	2.94	2.94
L. gasseri (128x)	97.9	97.9	97.9	96.4	100.0	97.9	1.81	1.85	1.86	0.66	1.85	1.85
S. mutans (134x)	100.0	100.0	100.0	99.3	99.9	100.0	2.03	2.03	2.05	1.67	1.70	0.68
H. pylori (477x)	100.0	100.0	100.0	99.3	100.0	12.3	1.66	0.95	1.68	1.04	1.30	-

(b) ZymoEven

	Reference Coverage					NGA50 (Mbp)				
	metaFlye	Canu	miniasm	wtdbg2	Flye	metaFlye	Canu	miniasm	wtdbg2	Flye
C. neoformans (10x)	83.7	80.8	38.1	40.3	0.1	0.03	0.03	-	-	-
S. cerevisiae (17x)	87.4	87.5	81.3	79.6	0.8	0.17	0.23	0.03	0.05	-
P. aeruginosa (155x)	100.0	100.0	100.0	89.3	100.0	6.78	6.77	4.45	0.70	6.78
E. coli (220x)	100.0	100.0	98.1	88.5	100.0	4.02	2.89	0.20	0.18	3.98
S. enterica (227x)	99.9	99.9	98.1	89.0	99.2	3.56	2.92	0.21	0.15	3.56
S. aureus (445x)	100.0	100.0	99.2	97.1	99.9	2.71	2.71	2.16	0.42	2.71
E. faecalis (464x)	100.0	100.0	99.9	97.3	100.0	2.82	2.16	2.85	0.30	2.83
B. subtilis (516x)	100.0	99.9	99.7	98.6	100.0	4.03	4.01	2.03	0.69	4.03
L. monocytogenes (525x)	100.0	100.0	99.4	98.6	100.0	2.10	2.98	2.86	1.64	2.98
L. fermentum (528x)	100.0	100.0	99.8	98.8	100.0	1.88	1.07	1.91	1.68	1.91

(c) ZymoLog

	Reference Coverage				NGA50 (Mbp)			
	metaFlye	Canu	miniasm	wtdbg2	metaFlye	Canu	miniasm	wtdbg2
C. neoformans (0.003x)	-	-	-	-	-	-	-	-
S. aureus (0.006x)	-	-	0.2	-	-	-	-	-
E. faecalis (0.08x)	-	-	0.1	0.4	-	-	-	-
L. fermentum (0.2x)	-	0.1	-	-	-	-	-	-
E. coli (2x)	36.5	13.6	0.4	14.5	-	-	-	-
S. enterica (2x)	34.8	10.4	-	11.5	-	-	-	-
S. cerevisiae (7x)	79.1	76.4	11.1	40.0	0.03	0.03	-	-
B. subtilis (37x)	99.7	100.0	98.8	98.9	1.23	3.20	0.74	0.77
P. aeruginosa (158x)	100.0	100.0	99.9	98.8	6.78	6.77	2.57	0.73
L. monocytogenes (3960x)	100.0	100.0	99.0	87.5	2.98	2.99	2.97	0.03

Figure 3. Per-species reference coverage and NGA50 statistics for the mock community datasets (HMP, ZymoEven GridION, ZymoLog GridION) computed using metaQUAST.

The read coverage for each species is given in the brackets after the species name. NGA50 values are not reported for assemblies with reference coverage below 50%. Blue and red colors correspond to the values higher and lower than the median, respectively. Flye failed to assemble the ZymoLog datasets due to poor *k*-mer indexing (Methods). Extended Data Figure 3 provides the base-pair quality analysis for the same datasets.

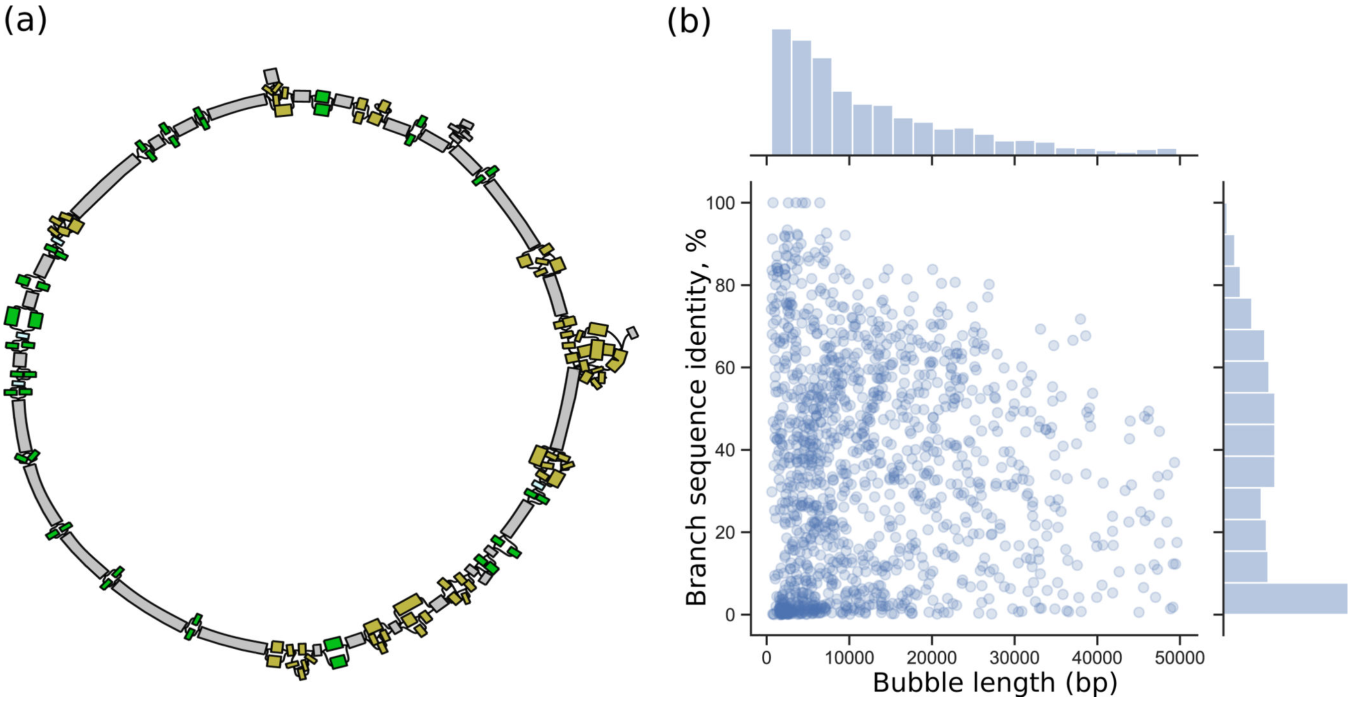


Figure 4. Information about strains in the sheep microbiome revealed by metaFlye.

(a) An assembly graph of a single connected component in the sheep microbiome dataset before strain collapsing (visualized using Bandage). The component represents a bacterial genome of the *Clostridia* class with 92% conserved marker completion (computed using CheckM). There are 20 simple bubbles (shown in green) and 10 superbubbles (shown in yellow) that account for 1.2 Mbp out of 2.4 Mbp long genome. (b) Distribution of length and branch sequence identities of 1141 bubbles (excluding loops and including roundabouts with only two edges) in the sheep microbiome assembly. The length is defined as the length of the longest branch in a simple bubble.

Table 1.

Assembly statistics for the mock community datasets.

Statistics were computed for contigs longer than 500 bp using metaQUAST 5.1.0rc1 with the default parameters. Misassembly counts are given for structural variations longer than 1 kbp (default value). The best value(s) in each category are highlighted in bold. NGAx is the NGx statistic computed for contigs that are broken at their misassembly breakpoints. Reference coverage is the percentage of the reference genome covered by assembled contigs. Sequence identity reported as a mean among all references. Two yeast genomes (*S. cerevisiae* and *C. neoformans*) did not contribute to the misassembly counts and sequence identity computation in all Zymo datasets. Miniasm contigs were polished using Racon. Flye did not assemble the ZymoLog datasets due to poor *k*-mer indexing (Methods). Canu and miniasm did not produce assemblies of the Zymo PromethION datasets due to large running time or memory requirements.

Dataset	Assembler	Assembly length, Mbp	Total reference coverage	Sequence identity	NGA50 (NGA25), kbp	Mis-assemblies	CPU hours
HMP 6.8 Gbp PacBio 19 bacterial references	metaFlye	66.4	**99.8%**	**99.9%**	**2,018**	72	45
	Flye	64.7	97.8%	**99.9%**	1,363	100	49
	Canu	**67.6**	99.7%	**99.9%**	1,854	105	756
	FALCON	60.0	90.3%	99.5%	764	116	150
	miniasm	66.6	99.6%	98.9%	1,863	**71**	11
	wtdbg2	65.6	98.7%	99.2%	675	101	**4**
ZymoEven GridION 14 Gbp ONT 8 bacterial & 2 yeast references	metaFlye	**63.8**	**95.7%**	**99.6%**	(3,559)	**7**	90
	Flye	31.1	51.5%	**99.6%**	**(3,562)**	10	105
	Canu	62.6	94.9%	99.4%	(2,920)	11	4,590
	miniasm	52.0	80.1%	99.3%	(2,032)	26	67
	wtdbg2	54.4	75.2%	99.3%	(329)	14	**5**
ZymoLog GridION 16 Gbp ONT 8 bacterial & 2 yeast references	metaFlye	**28.2**	**46.0%**	98.5%	(75)	40	112
	Flye	-	-	-	-	-	210
	Canu	25.3	41.9%	98.6%	**(81)**	**6**	38,800
	miniasm	15.6	26.4%	**99.2%**	(18)	43	299
	wtdbg2	23.2	33.7%	98.5%	-	24	**13**
ZymoEven PromethION 146 Gbp ONT 8 bacterial & 2 yeast references	metaFlye	**69.6**	**95.9%**	**99.5%**	**(3,013)**	**45**	1,410
	wtdgb2	25.8	41.8%	98.4%	(121)	50	**12**
ZymoLog PromethION 148 Gb ONT 8 bacterial & 2 yeast references	metaFlye	**37.7**	**57.7%**	**99.4%**	**(3,549)**	78	3,630
	wtdgb2	17.3	25.5%	97.4%	-	**52**	**16**

Table 2.

Long-read assemblies of real metagenomic datasets.

Human gut statistics are reported for the total of all separate assemblies of all samples. ORFs were clustered at 99% similarity. 16S rRNA genes were clustered into OTUs at 95% similarity. Matching 16S rRNA statistic reports the number of contigs with multiple 16S rRNA copies, where all copies are 97% similar (along with the total number of multi-copy contigs). CheckM statistics are reported for contigs with less than 5% contamination. Supplementary Tables 6-8 describes benchmarking of wtdbg2, miniasm, OPERA-MS, and Flye on the same datasets. Plasmids and viruses were identified in circular contigs shorter than 500 kbp using plasmidVerify and viralVerify, respectively.

Dataset	Sheep gut (this study)		Human gut (Bertrand et al. [6])		Cow rumen (Bickhart et al. [12])	
	metaFlye	**Canu**	**metaFlye**	**Canu**	**metaFlye**	**Canu**
Length in ctgs >10 kbp	1,454 Mbp	**1,540 Mbp**	**837 Mbp**	815 Mbp	**1,173 Mbp**	829 Mbp
Length in ctgs >100 kbp	**1,001 Mbp**	888 Mbp	**439 Mbp**	428 Mbp	**200 Mb**	60 Mbp
Length in ctgs >1 Mbp	**344 Mbp**	313 Mbp	**152 Mbp**	125 Mbp	**2 Mb**	0
Full-length ORFs	1,489,797	**1,569,187**	**969,005**	928,809	**1,316,090**	896,241
ORF clusters (99%)	**1,379,985**	1,350,267	**753,819**	704,087	**1,263,687**	811,419
16S rRNA genes	1,496	**1,679**	852	**1,091**	**539**	251
16S rRNA clusters (95%)	**263**	253	71	**91**	115	35
Contigs w/ matching 16S	**211 / 223**	198 / 203	**77 / 100**	76 / 116	**22 / 25**	8 / 8
CheckM >90% complete	**63**	49	**14**	12	0	0
CheckM >25% complete	**331**	291	**68**	60	**16**	6
Putative plasmids	**143**	12	**109**	63	**126**	51
Putative viruses	**284**	183	**49**	26	**249**	103
CPU hours	**450**	5,500	**1,020**	15,200	810	-