NAR Genomics and Bioinformatics, 2026, **8**, lqag062
https://doi.org/10.1093/nargab/lqag062
Pipelines and Workflows for Biological Data Analysis

OXFORD

LORA: a polymorphic multi-sample long read assembly pipeline

Dimitri Desvillechabrol [1,2], Rania Ouazahrou [1,2], Juliana Pipoli da Fonseca [3], Gerald F. Späth[3], Thomas Cokelaer [2,3,*]

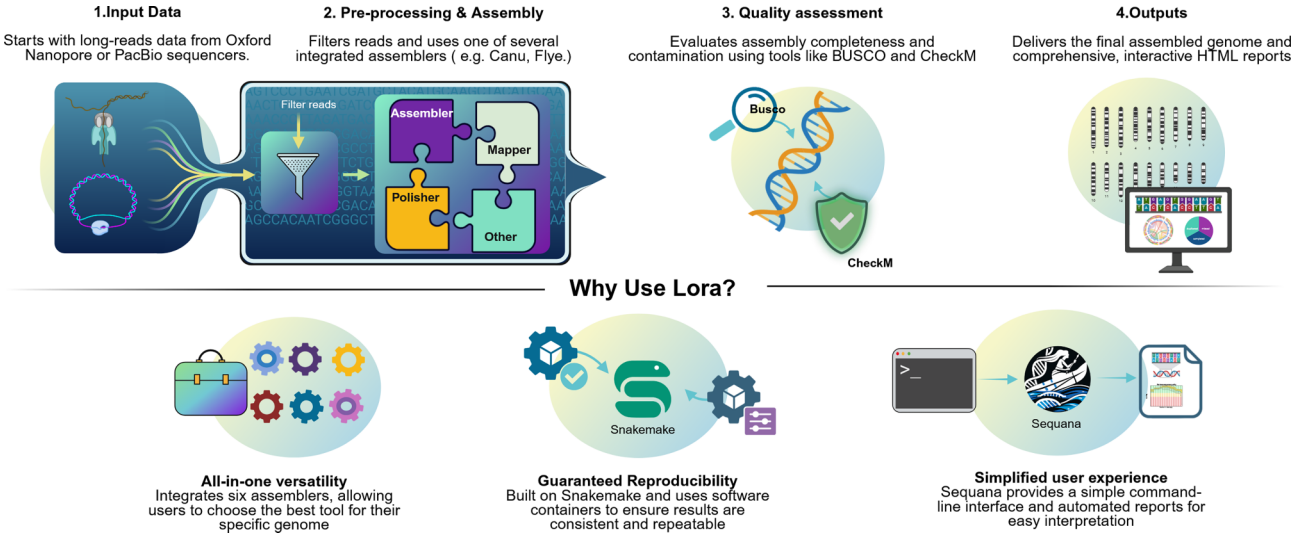
[1]Institut Pasteur, Université Paris Cité, Plate-forme Technologique Biomics, F-75015 Paris, France
[2]Institut Pasteur, Université Paris Cité, Bioinformatics and Biostatistics Hub, F-75015 Paris, France
[3]Institut Pasteur, INSERM 1347, Université Paris Cité, Parasitologie Moléculaire et Signalisation, F-75015 Paris, France

*To whom correspondence should be addressed. Email: thomas.cokelaer@pasteur.fr

Abstract

Genome assembly from long-read sequencing data has become a standard approach for resolving complex genomic regions and producing high-contiguity assemblies. However, the diversity of available assemblers, their varying performance across species, and the need for reproducible workflows present ongoing challenges. We developed LORA, an easy-to-use and reproducible application for assembling genomes from long-read data. LORA integrates several well-established assemblers, including Canu, HiFiasm, Flye, and Unicycler, as well as more recent tools such as Necat and Pecat. It is implemented as a Snakemake pipeline to parallelize tasks and support seamless execution on both local machines and computing clusters. LORA includes multiple quality assessment steps, interactive HTML reports for interpretation, BLAST-based taxonomic identification, and completeness evaluation. Together, these features provide users with a comprehensive view of assembly quality and potential problems. We illustrate the capabilities of LORA using datasets from bacterial genomes and unicellular eukaryotes, sequenced with both PacBio and Oxford Nanopore technologies, highlighting typical outcomes and common pitfalls encountered during long-read assemblies. LORA is distributed as part of the Sequana project, an open-source framework designed for reproducibility, maintainability, and straightforward deployment across computing environments.

Graphical abstract



Introduction

De novo genome assembly is a fundamental step in genomics [1], aiming to reconstruct complete genome sequences without the use of a reference. The advent of short-read sequencing platforms dramatically increased data throughput and enabled the development of new assembly algorithms [2, 3]. Nevertheless, assembling genomes from short reads remains challenging due to the sheer volume of data and the intrinsic structural complexity of many genomes. Although short-read technologies such as Illumina generate highly accu-

Received: January 21, 2026. **Revised:** April 1, 2026. **Accepted:** April 9, 2026
© The Author(s) 2026. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (https://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact reprints@oup.com for reprints and translation rights for reprints. All other permissions can be obtained through our RightsLink service via the Permissions link on the article page on our site—for further information please contact journals.permissions@oup.com.

rate sequences, their limited read lengths prevent the resolution of repetitive and high-complexity regions, often resulting in assemblies fragmented into many contigs.

The emergence of long-read sequencing technologies, including Oxford Nanopore Technologies (ONT) [4] and Pacific Biosciences (PacBio) [5], has transformed the field by enabling assemblies that span complex regions and frequently produce near-complete chromosomes in both prokaryotic and eukaryotic genomes [6–9]. PacBio and ONT rely on different sequencing chemistries, leading to distinct performance characteristics. PacBio offers two main read types: Continuous Long Reads (CLR) and High Fidelity (HiFi) reads generated by circular consensus sequencing. CLR reads typically reach accuracies of 87%–92% with N50 values of 30–60 kb, whereas HiFi reads exceed 99% accuracy with N50 values of 10–20 kb. Typical throughput ranges from 50 to 100 Gb for CLR and 15 to 30 Gb for HiFi [10, 11]. ONT Sequencing, in contrast, produces reads that can range from 10 to 200 kb, with accuracy up to 98% depending on the chemistry and base calling. A MinION flowcell generally yields 0.5–20 Gb, while the high-throughput PromethION platform can exceed 100 Gb [10].

As long-read sequencing has matured, many assemblers have been developed to exploit these data, enabling the reconstruction of high-quality genomes, especially for haploid species [12–14]. Benchmarking efforts have compared their performance across diverse organisms and datasets [15–17]. Still, not all genomes assemble easily. Bacterial genomes, for instance, can be grouped into three classes based on their repeat content [18]: class I genomes contain only small repeats (e.g. rDNA operons, 5–7 kb) and can often be assembled into a few contigs. Class II genomes harbor numerous mid-sized repeats (e.g. insertion sequences $\leq$ 7 kb), requiring higher coverage and longer reads for closure. Class III genomes contain large repeats ($\geq$7 kb), often phage-derived, such as tandem duplications and segmental duplications, which remain challenging even with long reads.

Diploid and polyploid genomes introduce additional complexity. Long reads improve haplotype reconstruction by spanning multiple heterozygous variants and enabling long-range phasing [19], a task largely intractable with short reads alone. However, long reads are not error-free: raw error rates of 5%–15% were reported for early ONT and PacBio CLR chemistries [20], and can obscure true heterozygous sites, complicate phasing, and lead to haplotype switch errors. More recent ONT chemistries, such as R10.4.1 with Kit 14, now achieve Phred Q20+ accuracy (below 1% error rate), dramatically reducing this concern for current datasets. Distinguishing heterozygosity from sequencing noise remains challenging, although modern basecalling and variant-calling models, such as DeepVariant [21], have improved diploid variant detection. Hybrid strategies that combine long reads with accurate short reads [22] can mitigate these issues and have enabled telomere-to-telomere assemblies of large eukaryotic genomes, including humans [23], albeit at substantial computational cost.

Because genomic architectures vary widely, assembly strategies must be tailored to the biological context: prokaryotic versus eukaryotic genomes, circular versus linear structures, plasmid presence, or levels of heterozygosity. For example, Flye [13] offers a dedicated plasmid-assembly module, while Canu [12] assemblies of circular genomes often require post-processing circularization tools. In heterozygous eukaryotes, phasing tools such as Whatshap [19] are ne eded to separate allelic sequences. Coverage depth also plays a critical role: insufficient coverage leads to fragmented assemblies, while excessively deep coverage can complicate graph resolution.

Assessing assembly quality is equally important. BUSCO [24, 25] evaluates completeness using conserved orthologs from OrthoDB, while CheckM [26] provides lineage-specific completeness and contamination estimates for prokaryotic assemblies. Metrics such as N50, alignment rates, and coverage profiles further characterize contiguity and accuracy, guiding downstream annotation and comparative analyses.

Although long-read assembly is now routine when sufficient coverage and read quality are available, outcomes remain highly dependent on input data characteristics and the choice of assembler. Comprehensive quality control, annotation, and reporting across multiple samples require substantial efforts. Reproducibility remains a major challenge in computational genomics, as highlighted by the ReproHackathons initiative [27], which advocates for workflow automation and containerization to overcome software and hardware heterogeneity.

To address these challenges, we introduce LORA (LOng Read Assembly), a reproducible, modular, and user-friendly Snakemake-based [28] pipeline developed within the Sequana project [29]. LORA provides an end-to-end solution for genome assembly from raw long-read data (FASTQ or BAM), supporting both PacBio and ONT platforms. It integrates multiple assemblers, optional polishing with short reads, and thorough quality assessment using BUSCO, CheckM, Quast, and sequana-coverage [30], with results consolidated in rich HTML reports. Rather than proposing a new assembler or performing an exhaustive benchmarking study, LORA offers a standardized, versatile, and reproducible workflow that simplifies tool selection, parameterization, and reporting across diverse genomic contexts.

In the following sections, we describe the LORA workflow (Materials and methods), evaluate its performance and usability on representative datasets (Results), and discuss its applicability and limitations across diverse genomic contexts (Discussion).

Materials and methods

Genome sequencing data and genome assembly data

To demonstrate the versatility of the LORA pipeline, we examined a variety of datasets covering both ONT and PacBio technologies, spanning different genomic complexities.

PacBio datasets

PacBio sequencing has evolved rapidly, producing different data types with distinct accuracy profiles. In this study, we first used FastQ files derived from the CLR data. More recent sequencers generate Consensus Circular Sequence (CCS) reads, including *HiFi* reads, obtained by performing multiple passes of the same molecule and applying a stringent quality filter (at least 10 passes, predicted accuracy of 99%, denoted *mp* and *rq*, respectively). While final CCS/HiFi reads are typically distributed as FASTQ files, the underlying data are stored in PacBio-specific BAM files. These BAM files differ from standard alignment BAMs: they encode raw subreads, per-base

qualities, polymerase metadata, and the information required to reconstruct CCS or HiFi sequences.

ONT datasets

For ONT, we considered several cases, including multi-sample datasets and data produced with different chemistries and basecalling models, such as the recent R10.4.1 model. In this study, we used data in FastQ format obtained after basecalling.

Genomic diversity

Regardless of the sequencing technology used, we analyzed genomes with varying repeat architecture and levels of biological complexity. Bacterial genomes can be classified into three repeat-defined classes according to [18]. Using 100 000 bacterial genomes, which were downloaded from the NCBI and are available via Zenodo entry zenodo.11519700, we updated this classification. Genomes with sizes below 1 Mb were removed, leaving 42 000 genomes for analysis. Repeats were identified with the `Repeats` class from the `Sequana` library [29]. Most genomes fall into class I and class III (96%), and these classes form the core of our long-read datasets. To broaden the applicability of LORA, we additionally considered a small eukaryotic genome, whose chromosomes can similarly be classified as class I and class III.

Datasets

Veillonella parvula—CLR PacBio data

As a representative of Class I genomes, we selected *Veillonella parvula* SKV38, whose GC content is 40% (see Supplementary Fig. S1) and contains only short repeats (below 1.5 kb; see Supplementary Fig. S2). Sequencing reads (ERR3958992) were obtained from the Sequence Read Archive (SRA) in FastQ format and published in [6]. The genome was sequenced on a PacBio Sequel II platform and assembled into a single contig of 2 146 482 bp (NCBI accession NZ_LR778174).

The dataset includes N = 338 310 reads with an average length of L = 9 kb and an N50 of 13 kb. The depth of coverage ($N \times L$/genome size) is therefore approximately equal to $1,500X$. To minimize misassemblies, the authors randomly subsampled 100 000 subreads, corresponding to $600X$ coverage. Assembly was performed with Canu v1.8, circularized using Circlator [31], and polished with Illumina data. The final assembly (RefSeq GCF_902810435.1) reports 98.36% completeness and 1.66% contamination according to CheckM [26]. The original BAM files are also available on Zenodo, enabling the reconstruction of CCS reads (DOI 10.5281/zenodo.13306683). The LORA workflow used for this case is illustrated in Supplementary Fig. S3.

Streptococcus gallolyticus subsp. *macedonicus*—HiFi PacBio data

As an example of a Class III genome, we analyzed *Streptococcus gallolyticuss* ubsp. *macedonicus* (SGM), which is characterized by a 40% GC content (Supplementary Fig. S4) and large repeats of 6 kb and 10 kb located in close proximity (see Supplementary Fig. S5). Sequencing reads (SRR24332397) were obtained from the SRA and published in [32] as part of the PRJNA94176 project. The final assembly (GCF_029866925.1) comprises a 2 210 410 bp chromosome and a 12 729 bp plasmid.

The dataset contains ∼2.5 million subreads in FASTQ format, generated on a PacBio Sequel II platform. In the original publication, the authors performed assembly using CCS reads generated with a minimum accuracy of 99% and at least three passes. However, these CCS reads are not available in the SRA, and the corresponding BAM file required to reproduce them was not publicly deposited.

To enable the generation of HiFi reads, we obtained permission from the authors to redistribute the original subreads BAM file. This file is now available on Zenodo (DOI: 10.5281/zenodo.17207091). Using this BAM file, we reconstructed HiFi CCS reads with the desired parameters (mp ≥10 passes, rq ≥ 99%). For convenience, both the HiFi BAM file and its corresponding FASTQ version are also provided in the Zenodo entry. The LORA workflow used for this case is illustrated in Supplementary Fig. S6.

Cyanobacteria—contamination Nanopore data

To showcase LORA's integrated quality assessment tools, we analyzed an unpublished dataset of Cyanobacteria that included contamination from two distinct strains showing a bimodal distribution of GC content (see Supplementary Fig. S7). Sequencing was performed on a GridION device using an R10.4.1 flow cell. Basecalling was performed using Dorado with the high-accuracy model v4.2.0.

The final assembly yielded two chromosomes of 4.41 Mb and 4.13 Mb. Repeat structure suggests a Class I classification for both genomes (Supplementary Figs S8 and S9). BLAST identified one chromosome as *Limnospira* and the other as *Geminocystis*. Data are available via Zenodo (DOI 10.5281/zenodo.17229238)

Bacteroides fragilis—multi-samples Nanopore data

To demonstrate LORA's capability to handle multiple samples, we reanalyzed data from Sydenham *et al.* [33], where several assemblers were benchmarked on *Bacteroides fragilis* isolates sequenced with PacBio, Nanopore, and Illumina technologies.

The reference genome (ASM2598v1, NCBI accession NC_003228.3) is categorized as Class I, with a GC content around 40% (Supplementary Fig. S11) and six 5 kb repeats and only a few 1–3 kb repeats (Supplementary Fig. S12). However, the study isolates contained much larger repeats (≥ 7 kb, up to 20 kb), spanning both Class I and III categories (see Fig. 2).

Nanopore sequencing was performed on a MinION device using R9.4 flow cells and the SQK-RPB004 ONT ligation kit; basecalling was performed with Albacore v2.3.3, as described in [33]. FastQ files are available via Zenodo entry zenodo.2677927). Additional data, including raw signal files (fast5 format), PacBio reads, and Illumina data, are accessible under BioProject accession numbers PRJNA525024, PRJNA244942, PRJNA244943, PRJNA244944, PRJNA253771, PRJNA254401, and PRJNA254455. The reference genome assembly mentioned above is available under NCBI accession number GCF000025985 and reports a completeness value of 99% and total length of 5 205 140 bp. A plasmid, pBF9342, is also available (NC_006873.1; 36 560 bp).

Leishmania—eukaryotes Nanopore

Finally, to evaluate LORA's applicability to eukaryotic genomes, we analyzed a *Leishmania* dataset from Almutairi

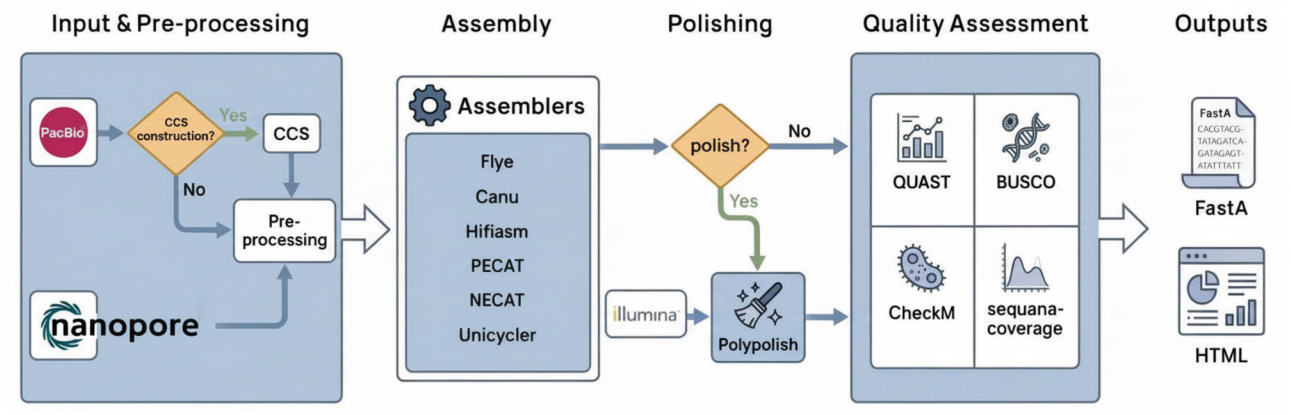


Figure 1. Overview of the LORA pipeline. Input and preprocessing accept PacBio (BAM/FastQ) and ONT (FastQ) data. It can automatically generate high-quality CCS/HiFi reads from PacBio BAM files. Assembly employs one of six integrated assemblers chosen by the user. Polishing is optional and can refine the assembly accuracy using short-read Illumina data. Quality assessment is made of a comprehensive suite of tools (BUSCO, Quast, CheckM, sequana-coverage) to evaluate contiguity, completeness, and contamination. The outputs deliver assembled contigs (FastA) and rich interactive HTML reports for easy interpretation.

et al. [34]. *Leishmania* genomes are diploid (often aneuploid), contain 34–36 chromosomes, and have a total genome size of ∼33 Mb, making them compact but structurally complex eukaryotic genomes.

We focused on the *Leishmania (Mundinia)* sp. Ghana strain, for which ONT and Illumina datasets are available (NCBI project PRJNA691536). Nanopore sequencing was performed on a MinION device using FLO-MIN106 flow cells [34]. Long-read FastQ files were used directly as deposited. The genome was assembled using a Snakemake-based workflow [34], yielding 125 contigs with an N50 of 961 565 bp (NCBI assembly GCA_017918215.1).

LORA pipeline

The analysis presented in this study was performed using LORA (version 1.0.1), a modular and extensible assembly pipeline. LORA relies on sequana_pipetools (version 1.0.3; sequana_pipetools), which provides the command-line interface for pipeline execution (see the "Discussion" section for details). The LORA source code and documentation are available within the Sequana project organization at github.com/sequana/lora. An overview of the LORA workflow is shown in Fig. 3. The main features of LORA are detailed in the "Results" section.

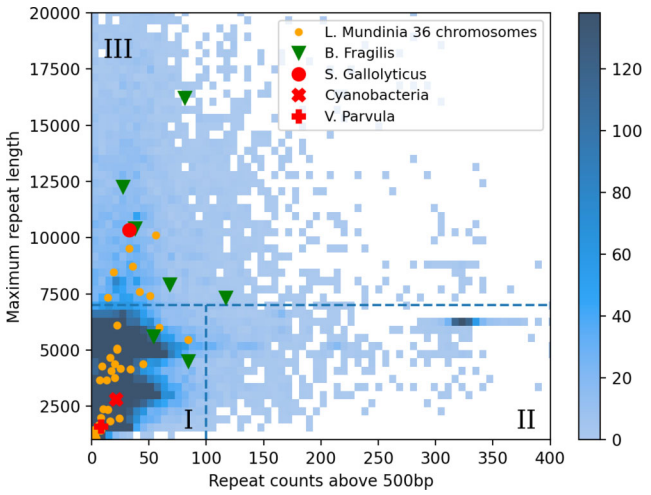


Figure 2. Repeat count versus maximum repeat length. This density plot includes 44 600 bacterial genomes of at least 1 Mb. For clarity, values above 20 kb for the repeat count are clipped, retaining 97% of the original dataset. Using a threshold of 7 kb for the maximum repeat length and a threshold of 100 for the number of repeats longer than 500 bp, three classes of genomes are defined. Classification and plot inspired by [18].

Results

Pipeline overview

LORA is a modular and reproducible Snakemake-based pipeline designed for long-read genome assembly. It is part of the Sequana project [29], which provides a comprehensive suite of next-generation sequencing pipelines covering genomics, transcriptomics, and both short- and long-read technologies.

At its core, LORA leverages Snakemake [28], a workflow management system that combines the readability of Python with a flexible domain-specific language. Snakemake ensures transparent rule definition, automatic handling of dependencies, efficient use of computational resources through concurrent execution of independent workflow tasks, and portability across different computing environments, from laptops to high-performance clusters. It is important to note that Snakemake's role is to schedule independent jobs concurrently; multithreading within individual assemblers is handled by the assemblers themselves and is controlled via their respective thread parameters in the configuration.

Beyond Snakemake, LORA benefits from the sequana_pipetools library, which provides a unified interface across all Sequana pipelines. This framework standardizes configuration handling, command-line options, and report generation. As a result, users familiar with one Sequana pipeline can readily adopt others, and developers can maintain consistency across the ecosystem. The user interface is deliberately kept simple: each pipeline can be launched with a single command, while advanced parameters can be fine-tuned via a configuration file compatible with Snakemake.

A simplified version of LORA is shown in Fig. 1. An example workflow, including task dependencies, is illustrated in

Fig. 3, which presents a rule graph for a specific configuration (one assembler, with Illumina polishing enabled). Tasks (or "rules" in Snakemake terminology) at the same level are essentially independent and can be executed concurrently.

Moreover, the pipeline has a polymorphic design, adapting its structure based on the user's input data and chosen options. For example, Fig. 3 includes optional tasks such as polishing with Illumina data or performing quality assessments. However, depending on user's choice, the final workflow can take several forms (see Supplementary Figs S3, S6, S10, S13, and S14).

In the following sections, we provide a detailed description of LORA's components, including supported assemblers, quality assessment modules, and reporting features.

Reporting

Once LORA completes its analysis, all results are organized in a clear directory tree. Each sample has its own subdirectory, which in turn contains folders for each task (e.g. preprocessing, assembly, polishing). To make it easy for users to explore the outputs, LORA generates a set of HTML reports. The first report is produced with MultiQC [35], providing multi-sample summaries for tools such as fastp, BUSCO, sequana_coverage, and others. LORA also provides its own entry point (summary.html), which links to the MultiQC report, the pipeline configuration file, its dependencies, an overview of the workflow, and a summary table for each sample, including key metrics such as the number of contigs, and links to Quast reports. The summary page also links to a more complete report that includes links to individual contig coverage plots. When BLAST analysis is enabled, the report also includes information on the best BLAST hits for each contig. If enabled, it also contains the BUSCO and Checkm metrics.

Input data

LORA accepts long-read sequencing data in standard FASTQ format, supporting both PacBio and ONT datasets. For ONT, only base-called FASTQ files are supported; raw signal formats such as pod5 or fast5 are not handled directly. PacBio data may be provided in FastQ format or BAM files. Although BAM files can be converted to FASTQ with third-party tools (e.g. using BioConvert [36]), LORA natively supports them by converting them into FastQ files.

PacBio sequencing generates subreads, i.e. single passes over a DNA molecule, which typically exhibit an error rate of 10%–15%. To reduce these errors, PacBio can perform multiple passes over the same DNA molecule and generate a consensus sequence (CCS) known as HiFi reads. HiFi data usually require at least ten passes and can achieve an accuracy of 99%. When users provide original subreads in BAM format, LORA can automatically generate HiFi or other CCS reads. This step is efficiently incorporated into the workflow by splitting the input BAM files, computing consensus sequences, and merging the resulting data back into a single BAM file.

Preprocessing

Once the input data are available as FastQ file, provided directly by the user or through the building of the CCS data (PacBio only), LORA processes the data using FastP [37], allowing for quality control and filtering. By default, reads shorter than 1000 bp are filtered out, but users can customize this threshold and apply additional FastP options, such as quality trimming.

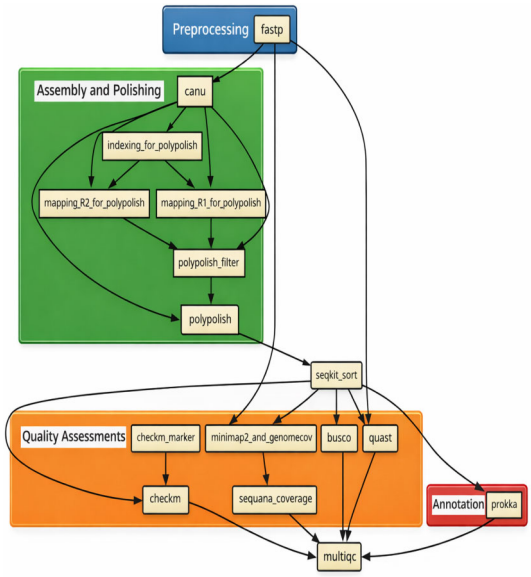


Figure 3. Example of LORA workflow. This workflow begins with input FASTQ data (blue box), followed by assembly with Canu and polishing using Illumina reads (green box). The resulting contigs are then subjected to quality assessments (orange box) and annotation (red box). Final results consist of HTML reports and contigs in FastA format.

Assemblers included

The pipeline integrates several assemblers that have different strengths depending on the nature and quality of the input data.

Canu

The Canu assembler [12] is known for its ability to manage highly erroneous reads and produce high-quality assemblies. Canu implements an overlap–layout–consensus (OLC) approach first described by Rodger Staden in 1979. Canu includes dedicated modules for read correction, trimming, and assembly. A genome size estimate can be provided by the user, although this parameter is not directly enforced during contig construction. Circularization is not included and should be performed with other tools such as Circlator [31] to close circular chromosomes or plasmids. It does not explicitly model diploidy; heterozygous regions are typically collapsed into a consensus sequence. Note that active development of Canu has ceased; however, because LORA distributes all tools as versioned Apptainer containers through the Damona project, Canu remains fully reproducible and functional within the pipeline. Its inclusion is retained for benchmarking continuity and backward compatibility with existing workflows.

Flye

Flye assembler [13] assembles genomes using a repeat graph, which accommodates high sequencing error rates and captures the genome's repeat structure. This approach enables the recovery of long contigs even from noisy long-read data. Flye can optionally circularize replicons internally, making explicit post-processing unnecessary in many cases. For diploid genomes, Flye generally produces a collapsed mosaic haplotype rather than fully phased sequences, though the original

assembly graph can be retained for downstream haplotype analysis or inspection.

Hifiasm

LORA also includes the Hifiasm [38] assembler that is optimized for PacBio HiFi reads and produces high-quality, haplotype-resolved assemblies. It begins with an all-versus-all overlap of reads, followed by several rounds of error correction, then constructs a string graph where bubbles represent heterozygous alleles. Depending on user options and parental data, Hifiasm can create primary/alternate contigs or fully phased haplotypes. Circularization is not an explicit feature, as Hifiasm is primarily targeted at linear chromosomes and phased eukaryotic assemblies.

NECAT

NECAT [39] is a de novo assembler designed primarily for ONT, though it is compatible with PacBio CLR data. It adopts an OLC strategy with a progressive error-correction scheme: low-error reads are corrected first, followed by high-error reads. This adaptive procedure accelerates correction while preserving accuracy. During assembly, NECAT constructs contigs from the corrected reads and subsequently bridges them with raw reads to improve continuity. An adaptive selection mechanism is used both to identify high-quality supporting reads for each template during correction and to retain the most reliable overlaps in the layout stage. NECAT can assemble bacterial, fungal, and moderately sized eukaryotic genomes on standard hardware. It does not include an internal circularization step, and heterozygous regions are generally collapsed rather than phased. Compared with "correct-then-assemble" tools such as Canu, NECAT is reported to be 2.5–258× faster while producing assemblies of similar quality; relative to "assemble-then-correct" assemblers such as Flye, NECAT achieves fewer misassemblies on complex genomes [39].

PECAT

Another assembler included in LORA is PECAT [40] (Phased Error Correction and Assembly Tool). It is also specifically designed for long, noisy reads (e.g. Nanopore or PacBio CLR) to reconstruct diploid genomes with haplotype-aware accuracy. It follows a correct-then-assemble approach: a haplotype-aware error correction step that preserves heterozygous alleles followed by two rounds of string-graph-based assembly. The second step uses SNP-calling components to filter inconsistent overlaps and reconstruct haplotype-specific contigs, either in "primary/alternate" or in "dual assembly" format. PECAT achieves very high continuity (e.g. large N50) and compares favorably to other diploid-capable assemblers. It is efficient in CPU time and memory, relative to more classical correct-then-assemble pipelines, and demonstrates good performance even with noisy reads, especially as coverage and read quality increase. PECAT does not include built-in circularization, as its focus is on accurate reconstruction of phased contigs in diploid or polyploid contexts.

Unicycler

Unicycler [14] is a *de novo* genome assembler developed primarily for bacterial genomes, supporting short-read, long-read, and hybrid assembly modes. It integrates the short-read assembler SPAdes [3] with a long-read bridging algorithm, enabling the resolution of repetitive regions and the closure of circular chromosomes or plasmids when long reads are available. Although Unicycler can assemble Illumina data alone, long reads alone, or a combination of both, LORA currently supports only the long-read mode. In this mode, Unicycler employs `minimap` and `miniasm` to construct an initial assembly, which is uncorrected and therefore retains an error rate similar to that of the input reads. The bundled version of miniasm is modified to better assemble circular replicons into circular string graphs. Unicycler also incorporates polishing steps (e.g. Racon) to refine consensus accuracy and improve genome continuity. Owing to its graph-based architecture, Unicycler is particularly well suited to small, low-complexity genomes but may be less efficient for large or highly heterozygous eukaryotic datasets [14].

Post processing

Circularization

After the assembly step, bacterial genomes and plasmids often require an additional circularization step to represent their natural topology. Assemblers may output a linear contig even when the underlying molecule is circular, leaving duplicated ends or small overlaps. To address this, LORA includes a circularization module based on Circlator [31] that scans contig termini for significant overlap, trims redundant regions, and reorders the sequence so that the origin of replication or another user-defined position appears at the start. This procedure ensures that circular chromosomes and plasmids are accurately represented as single, gap-free sequences, facilitating downstream analyses such as annotation and comparative genomics. Circlator is currently not maintained (sanger-pathogens/circlator). A fork was created within the Sequana project to fix several bugs. This version has been containerized and is distributed through the Damona project as a reproducible container.

Scaffolding

Assemblers generally output a set of contigs, representing contiguous stretches of sequence but lacking information about their relative order and orientation. When a reliable reference genome is available, these contigs can then be aligned, generating a scaffold. This alignment organizes the contigs into the correct order and orientation, while introducing placeholder gaps (denoted by N bases) for regions that remain unresolved. LORA integrates the RagTag suite [41] to perform this scaffolding step efficiently, mapping contigs to the reference and generating a scaffolded assembly that is ready for downstream analyses.

Quality assessments

Quality assessment is a critical step in evaluating the accuracy, completeness, and reliability of a genome assembly. Several key metrics are commonly employed to assess assembly quality.

N50

The N50 metric is a widely used metric to assess genome assembly contiguity. It is defined as the length N such that 50% of the total assembly length is contained in contigs (or scaffolds) of length greater than or equal to N. To calculate it, contigs are first ordered from longest to shortest, and their lengths are cumulatively summed until half of the total assembly size is reached. The length of the contig at this threshold

is reported as the N50 value. While N50 is most informative for assemblies with many contigs, it can be less meaningful for assemblies generated from long-read data with only a few contigs. In LORA, Quast [42] is used to compute the N50.

Alignment rates

This measures the proportion of reads from the original long-read sequencing data that successfully map to the assembled contigs. High alignment rates indicate that a large fraction of the sequencing data has been incorporated correctly into the assembly. In LORA, Quast [42] is used to compute the alignment rate.

Completeness

Another useful metric is completeness. It reflects the proportion of a genome successfully recovered in the assembly. It is typically estimated by identifying conserved, single-copy marker genes that are expected to be present in all members of a particular taxonomic group. LORA integrates BUSCO (Benchmarking Universal Single-Copy Orthologs) [24], which assesses assembly completeness by comparing the genome to a set of highly conserved, single-copy genes from OrthoDB [25]. The analyses presented in this study were performed with BUSCO v6.0.0 using OrthoDB v12 (odb12) lineage databases [25]. A high BUSCO completeness score indicates that most essential genes have been successfully captured. LORA also includes CheckM [26], which provides a completeness estimate for bacterial genomes and offers finer resolution across sub-taxonomic groups.

Contamination

This refers to the presence of sequences from other organisms within the assembled genome. This can occur when metagenomic sequences from different species are incorrectly combined during assembly. Both BUSCO and CheckM provide contamination estimates. BUSCO identifies duplicated marker genes that may indicate contamination from closely related strains. CheckM distinguishes between two types of contamination: (i) fragments originating from multiple strains of the same species, and (ii) fragments from more distantly related taxa (heterogeneity). High heterogeneity can suggest the presence of sequences from different strains or subspecies.

Genome Coverage

Depth of coverage along the genome is useful to evaluate how much of the target genome is represented in the assembly. Higher coverage generally correlates with higher completeness. In LORA, genome coverage is computed using `sequana_coverage` [30], which maps reads to contigs, estimates coverage trends, and provides visualizations for inspecting contigs, detecting copy number variations (CNVs), and identifying coverage drops.

Collectively, these quality assessment metrics offer a comprehensive view of assembly performance, helping researchers gauge the reliability and utility of the genome assembly. A high-quality assembly should exhibit high contiguity, completeness, accuracy, and low contamination, making it a valuable resource for downstream genomics and bioinformatics analyses.

Annotation option

Prokka

LORA is dedicated to assembly. However, in the context of bacterial genomes, we can use Prokka [43] to annotate the final contigs. Currently, no annotation is planned for eukaryotes, which usually requires specific domain knowledge.

Blast

LORA optionally supports the annotation of assembled contigs through BLAST searches against user-supplied reference databases. Because the size and scope of BLAST databases vary, depending on the intended application, LORA does not provide or distribute reference datasets. Instead, users are responsible for preparing and maintaining a valid local BLAST database prior to execution. When such a database is available and properly configured, LORA performs BLAST queries for each contig and incorporates the most relevant hits into its HTML report, thereby facilitating the inspection of putative annotations and improving the interpretability of the assembly.

LORA pipeline use cases across diverse genomes

In this section, we illustrate the versatility and flexible design of the LORA pipeline. Figure 3 shows a representative workflow; however, the actual execution may vary depending on the user's setup, input data, selected mode (bacterial or eukaryotic), or choice of assembler. The datasets used in the examples below are described in the "Materials and methods" section and summarized in Table 1. They span different sequencing technologies and varying data quality over time.

Veillonella parvula—CLR PacBio data

This case evaluates LORA's ability to handle noisy CLR reads and compare assembler sensitivity to read quality. The original dataset, composed of subread files in BAM format, can be used directly by the pipeline, which automatically converts them to FASTQ files. Alternatively, the BAM files may be used to build circular consensus sequences (CCS) for downstream assembly.

The genome under study (*Veillonella parvula*) is classified as class I and should not be difficult to assemble. Nevertheless, the dataset considered was sequenced on a PacBio Sequel II platform with an older chemistry version (v1.2) without multiplexing. As a result, the data yield was large (Table 1), but each molecule was sequenced only a few times. Using LORA, we first attempted to generate HiFi reads from the subreads; however, only a limited number of subreads met the HiFi quality criteria. In the original study [6], the authors used a subsample of 100 000 raw subreads (coverage 400*X*) without constructing CCS reads. Following this strategy, we performed two experiments: (i) assembling a random subsample of 100 000 subreads and (ii) generating CCS reads by disabling filtering on the number of passes (setting the minimum to 0) and requiring an accuracy threshold of 0.7, yielding 175 000 CCS reads.

For both datasets, we ran all 6 assemblers available in LORA (Section Assemblers included) with and without the circularization option, resulting in 32 individual pipeline executions. Default parameters were used, and the `–mode bacteria` option was enabled to run BUSCO (bacterial lineage) and CheckM, using *Veillonella parvula* as the reference

Table 1. Data Summary

cases	genome size (Mb)	GC%	class	data type	N	$\tilde{L}$(kb)	N50 (kb)	DOC
Veillonella parvula	2.1	38.7	I	PB-subreads	338 310	9.	13	1400
				PB-CCS[a]	192 532	8.8	12.8	800
Streptococcus gallolyticus	2.2	37.7	III	PB-subreads	2 552 698	8.9	11.3	10 000
				PB – HiFi[b]	34 013	7.3	7.8	113
Bacteroides fragilis	∼ 5.2	43.1	I, III	ONT R9.4.1	20 000–304 967	[5.9–8]	[12–15.7]	[23–369]
Cyanobacterium	2 × 4.5[c]	56.8	I	ONT R10.4.1	1 572 000	1.9	2.6	695
Leishmania mundinia	35.9	59.5	I, II, III	ONT R9.4.1	770 984	7.0	19.5	161

Notes: [a] Default parameters of the LORA pipeline were used (rq > 0.7, mp ≥ 0). [b] HiFi reads were generated with rq > 0.99 and mp ≥ 10. [c] Due to contamination, two cyanobacterial genomes of ∼4.5 Mb each were present.
First column lists the species used in each test case. Column 2 reports the approximate genome length, followed by the expected GC content and the genome's complexity class. For the *Leishmania* dataset, the complexity class was determined for each of the 36 chromosomes, which fall into three distinct categories. Additional information is provided for each dataset, including the sequencing data type, from either PacBio or Nanopore (for PacBio, datasets may consist of raw reads, CCS reads, or HiFi reads). The table then lists the number of reads in the raw or processed data, the mean read length ($\tilde{L}$), the N50, and the expected depth of coverage (DOC; $N \times L$/genome size).

species. Assembly and completeness results are shown in Figs 4–7.

Using the subsampled subreads, most assemblers achieved a good assembly, as shown in Fig. 4. Canu produced a single contig when the circularization option was enabled; without circularization, the same contig was obtained, but an additional small contig was reported (not supported by coverage). CheckM completeness reached 98.5%. Flye produced one complete contig with or without circularization. Flye correctly identified the origin of replication even without explicit circularization, indicating robust internal handling of circular genomes (CheckM: 98.4%). Hifiasm failed on this dataset, likely due to the high error rate in the raw reads. Necat generated a single chromosome, but quality control exposed the low quality of the assembly (BUSCO: 9 missing and 21 fragmented genes; CheckM: 93%), with several local coverage drops. Pecat yielded a single chromosome, with similar issues (8 missing genes and 19 fragmented BUSCO genes). Circularization slightly improved CheckM (≈ 95%) but not the BUSCO scores. Unicycler performed poorly, producing seven contigs, a CheckM score of 97.5%, and only 76 complete BUSCO genes.

When using CCS reads, coverage increased, and read quality improved (≈175 000 CCS reads). Canu produced results similar to those obtained from raw reads, consistent with its internal correction step (CheckM: 98.4%). Flye again achieved an excellent assembly (CheckM: 98.4%, BUSCO: 112/124), with no benefit from circularization. Hifiasm finished its tasks but remained unsuitable, although enabling circularization reduced the number of contigs from seven to two but introduced substantial coverage drops (CheckM: 95%). Necat assembled a complete chromosome but retained low CheckM completeness (93%) and fragmented BUSCO genes. Pecat also reconstructed the chromosome with the correct length but yielded CheckM completeness of only 94.5% and irregular coverage. Unicycler produced seven contigs with a total size exceeding the expected genome length by ≈ 20%.

In summary, Flye and Canu consistently produced high-quality assemblies, with Flye generally yielding the most accurate and stable results, independent of whether circularization was applied. In contrast, Hifiasm was sensitive to read quality, and Unicycler tended to over-fragment the assembly. Necat and Pecat were able to generate single-chromosome assemblies, but with reduced completeness and irregular coverage. These results demonstrate that LORA enables users to systematically benchmark assemblers and parameter choices for optimal genome reconstruction.

SGM—HiFi PacBio data

Here, we test LORA's performance on repeat-rich, high-quality HiFi datasets. We considered a bacterium classified as class III due to a 10 kb repeat and several 6 kb repeats, notably near the origin of replication (Supplementary Fig. S4). Starting from the raw data, we set the pipeline to build the PacBio HiFi CCS using the `-pacbio-build-ccs` option. This is different from [32], where CCS reads were generated with ≥ 3 passes and an accuracy above 99%.

After running the pipeline for each assembler, both with and without circularization, all assemblers completed the assembly. A summary of the number of contigs obtained for each assembler is shown in Fig. 8, and all BUSCO scores are provided in Fig. 9.

Combined with a circularization step, Canu produced an almost perfect assembly comprising a circular chromosome and plasmid. Canu also assembled six small artifactual contigs that can be easily discarded. The plasmid showed only minor local drops in coverage when the dataset was mapped to the assembly. If we omit the circularization step, the chromosome is split into two contigs and the plasmid is incompletely resolved.

Flye consistently returned two high-quality contigs representing the chromosome and plasmid. The circularization step mainly adjusted the origin of replication without altering continuity.

With circularization, Hifiasm yielded a complete chromosome and plasmid, plus a single unsupported contig that can be ignored. Without the circularization step, the plasmid assembly was inaccurate (excess length).

Assemblies produced by NECAT, irrespective of the circularization step, were incomplete: the largest contig covered ∼80% of the chromosome, accompanied by a 70 kb fragment and a misassembled plasmid.

PECAT delivered results comparable to Flye. The circularization step slightly reduced overall quality, introducing minor coverage dips.

With and without the circularization step, Unicycler generated fragmented, yet complete assemblies (7–11 contigs) and the plasmid was recovered with high precision.

Overall, Flye and PECAT provided the most straightforward and accurate reconstructions, while Canu (with circu-

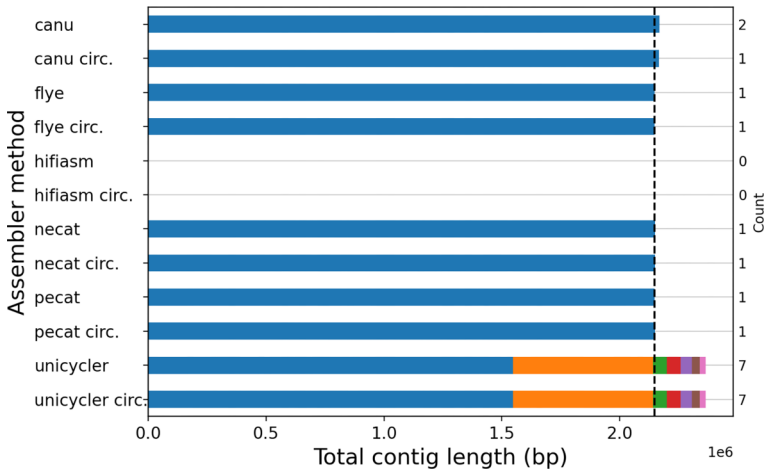


Figure 4. Assembly results for *Veillonella parvula* using PacBio raw reads. Each horizontal bar represents the total size of the assembled contigs (*x*-axis), with the corresponding number of contigs shown on the right *y*-axis. The assembler used, along with whether circularization was applied, is indicated on the left *y*-axis. The vertical dashed line indicates the expected total size. Overall, most assemblers successfully completed the assembly, with the exception of Hifiasm, which failed to finish the assembly task. Unicycler produced seven contigs. Colors are used solely for visual distinction and carry no specific meaning.

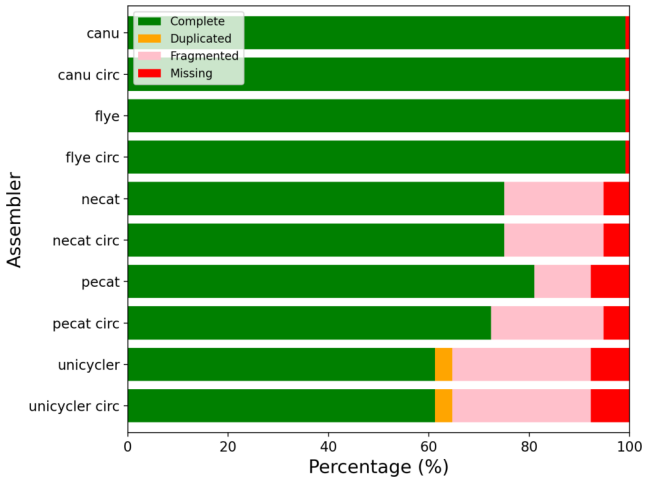


Figure 5. BUSCO scores obtained for the *Veillonella* case (subreads). The *y*-axis shows the assemblers used for this analysis and the *x*-axis shows the Busco score. While Canu and Flye achieved high completeness (bacterial lineage), others exhibit significant proportion of fragmented or missing genes.

larization) also performed well after filtering small contigs. Hifiasm results were also quite good regarding contiguity and completeness because of the high quality of the data set. NECAT and Unicycler were clearly less suitable for this genome, and circularization had tool-specific effects, generally improving assembly orientation but occasionally affecting coverage uniformity.

Cyanobacterium—Nanopore R10.4.1 with contamination

This case demonstrates LORA's capacity to disentangle mixed genomes and identify contamination. We use a recent high-yield Oxford Nanopore dataset. Although the sequencing experiment targeted a single cyanobacterium, contamination was present, resulting in two species (estimated genome sizes 4.1 Mb and 4.4 Mb) together with two plasmids. This dataset is particularly interesting because it enables an assessment of how assemblers available in LORA behave in complex, mixed samples. A summary of the number of contigs obtained for each assembler is shown in Fig. 10, and all BUSCO scores are provided in Fig. 11.

Canu, without circularization, successfully reconstructed the two bacterial chromosomes with uniform coverage. CheckM reported 99.2% completeness but also a contamination level of 90%, while BUSCO revealed that over half of the genes were duplicated, reflecting the presence of both species. Plasmids were detected but not fully assembled. Enabling circularization in Canu yielded similar results, with improvements in fully closing the circular chromosomes.

Flye produced highly contiguous assemblies in both modes. Without circularization, it recovered the two chromosomes with 99.5% completeness, alongside four additional contigs: a correctly assembled 35 kb plasmid, a small 12 kb plasmid with proper 400× coverage, a 14 kb fragment with abnormally high coverage (8000×), and another low-quality sequence. Applying Flye's circularization step did not change these results, indicating that internal handling of circular elements is already effective.

Necat (with or without circularization) assembled one of the genomes as a single 4.1 Mb contig, while the second was split into two pieces. Completeness was slightly lower than Flye (≈ 99%), and coverage plots showed localized drops. Both plasmids were retrieved, along with the same spurious high-coverage contig observed with Flye.

Pecat reconstructed the two chromosomes and plasmids when run without circularization, achieving 98.5% completeness, though some coverage dips remained. However, Pecat with circularization failed to complete. Unicycler was unable to finish (memory usage exceeded 64 GB), and Hifiasm generated an excessive number of contigs; its circularization step did not converge, likely because of the fragmented assembly graph.

Overall, Flye offered the most balanced outcome for this contaminated Nanopore dataset, providing near-complete assemblies for both cyanobacteria and the plasmids, with minimal artifacts. Canu also performed well, while Necat and Pecat produced usable results but with

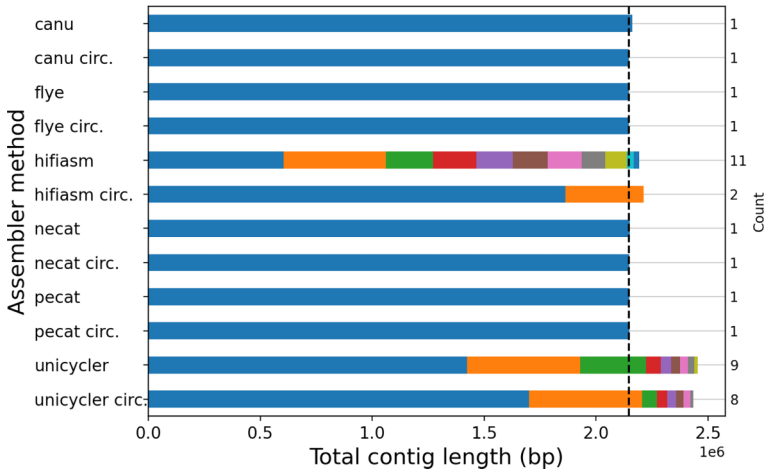


Figure 6. Assembly results for *Veillonella parvula* using PacBio CCS reads (see text for details). Each horizontal bar represents the total size of the assembled contigs (*x*-axis), with the corresponding number of contigs shown on the right *y*-axis. The assembler used, along with whether circularization was applied, is indicated on the left *y*-axis. The vertical dashed line indicates the expected total size. Overall, most assemblers successfully completed the assembly, with the exception of Hifiasm and Unicycler, which generated ∼10 contigs. Colors are used for distinctions of the contigs.

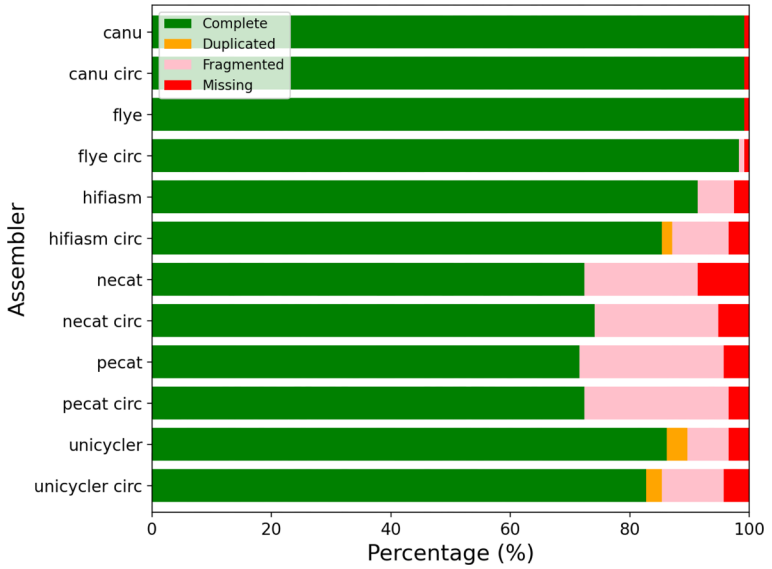


Figure 7. BUSCO scores obtained for the *Veillonella* case (CCS). On the *y*-axis, the assemblers used for this analysis; on the *x*-axis, the Busco score. While Canu and Flye achieved high completeness (bacterial lineage), others exhibit significant proportion of fragmented or missing genes.

slightly reduced completeness. Assemblers such as Unicycler and Hifiasm were not suited for this challenging mixed sample.

Isolates of *Bacteroides fragilis*

This case illustrates LORA's multi-sample management and integrated polishing features. We use a Bacteroides dataset consisting of seven samples sequenced with both ONT, PacBio CLR, and Illumina short reads. This dataset was used to demonstrate (i) the ability of LORA to process multiple samples in a single run and (ii) the integration of polishing steps based on short-read data. To handle multi-samples, users simply need to indicate where to find the input files (`–input-directory`) and a pattern (default uses the wildcard `*.fastq.gz`). For polishing, LORA incorporates the Polypolish tool [44]. Similarly to the long read data, users need to indicate where to find the FastQ files (`–polypolish-input-directory`). LORA handles single-end and paired-end data sets.

Assemblies were first generated from ONT data using Flye. Across the seven samples, assemblies ranged from 3 to 11 contigs with a total length of 5.4 Mb ± 0.15 Mb. Coverage varied from 20× to 200×, with lower coverage samples producing a higher number of contigs. Some assemblies included circularized replicons, such as sample S01 (a 5.5 Mb chromosome and a 55 kb plasmid) and nBF02 (three circular contigs corresponding to the chromosome and two plasmids). Overall, the assemblies were structurally sound. However, quality metrics revealed limitations: CheckM completeness was 81%, and BUSCO scores (Fig. 12) indicated fragmentation (41%) and missing genes (15%), consistent across all seven samples.

Assemblies were then repeated with PacBio CLR data at 100× coverage. In this case, final assemblies were more fragmented, with 10–40 contigs, with a total length of 5.5 Mb ±

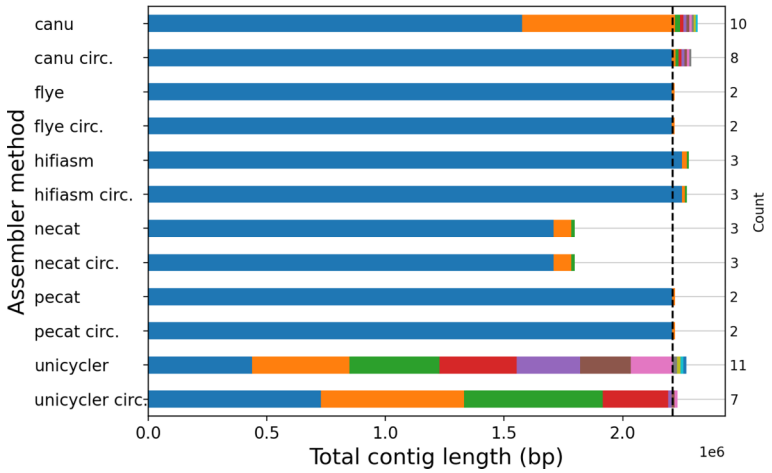


Figure 8. SGM assembly results using HiFi CCS reads with 110× coverage for SGM. Each bar corresponds to the number of contigs obtained (*y*-axis, right) and its size (*x*-axis), depending on the assembler used (*y*-axis, left). The expected genome consists of one full chromosome and one plasmid. All assemblies successfully recovered the plasmid. Canu generated 10 contigs and Unicycler generated 11 contigs, each one represented by its respective color on the graph.

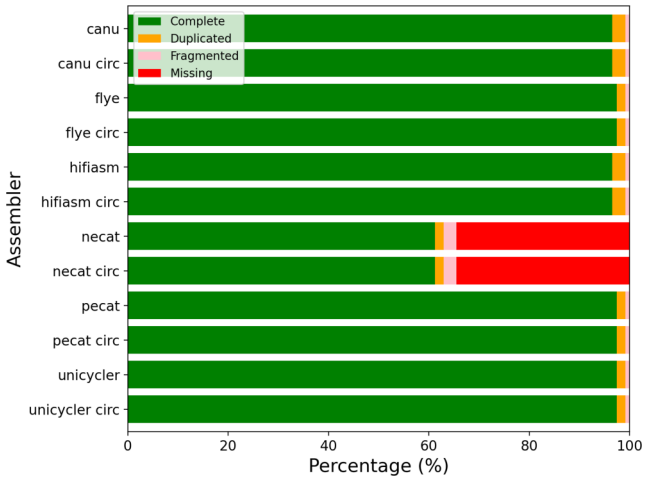


Figure 9. BUSCO scores obtained for the SGM case. On the *y*-axis, the assemblers used for this analysis; on the *x*-axis, the Busco score. Except for NECAT, all assemblers achieved a high completeness on the bacterial lineage.

0.2 Mb. However, the consensus accuracy was higher. Here, CheckM completeness reached 99% and BUSCO completeness was 98.5%, a striking contrast to the 43% observed with ONT-only assemblies.

To assess the benefit of ONT-based polishing, we applied Medaka [45] to the Flye assemblies. Medaka improved BUSCO completeness from 41% (raw Nanopore) to 61%, while reducing fragmented and missing genes, with contiguity remaining unchanged. Although this represents a meaningful gain, BUSCO completeness remained well below the PacBio level.

Finally, to demonstrate LORA's integrated short-read polishing functionality, we combined ONT assemblies with Illumina paired-end reads. Two polishing modes were tested: one using paired reads and one using only the first mate. In both cases, the overall contiguity of assemblies remained unchanged, but polishing improved gene completeness: BUSCO completeness increased to 98.2% and CheckM completeness to 99%, demonstrating the effectiveness of short-read polishing in correcting long-read assemblies.

Applying LORA on a small eukaryote genome from *Leishmania*

Finally, we assess LORA's applicability to small diploid eukaryotic genomes. We considered the genome of *Leishmania mundinia*, a kinetoplastid parasite with a complex genome. This organism is typically diploid, although not all chromosomes are strictly diploid, and its genome comprises 36 chromosomes totaling ∼33 Mb. We used low-coverage Oxford Nanopore long reads as input as described in the "Materials and methods" section.

Although LORA ships with assemblers designed primarily for bacterial genomes (e.g. Unicycler), it also integrates tools capable of managing more complex assemblies, including diploid eukaryotes. The aim of this case is not to benchmark assemblers exhaustively but to demonstrate that LORA supports a variety of tools suitable for eukaryotic genomes.

A summary of the number of contigs obtained for each assembler is shown in Fig. 13, and all BUSCO scores are provided in Fig. 14.

Among the tested assemblers, Canu failed to produce an assembly (>48 h computation) and Hifiasm failed to produce meaningful contigs (200 contigs of length below 3 kb). Flye generated 234 contigs spanning 36 Mb, with the largest contig of 1.5 Mb. In contrast, NECAT achieved better contiguity, producing 80 contigs, including a 2.7 Mb contig corresponding to the complete chromosome 36. PECAT performed similarly well, yielding 62 contigs for a total size of 38.6 Mb, with the same chromosome recovered as a single contig.

LORA also provides an optional scaffolding step using a known reference. When applied to NECAT and PECAT assemblies, this feature reconstructed all 36 chromosomes with only 16 and 14 gaps, respectively. Both NECAT and PECAT assemblies reached an N50 of 800 kb, compared to 500 kb for Flye. However, NECAT and PECAT showed slightly more fragmented BUSCO genes (+3 relative to Flye) and an increase of 3 Mb in assembled sequence, suggesting a trade-off between contiguity and assembly accuracy.

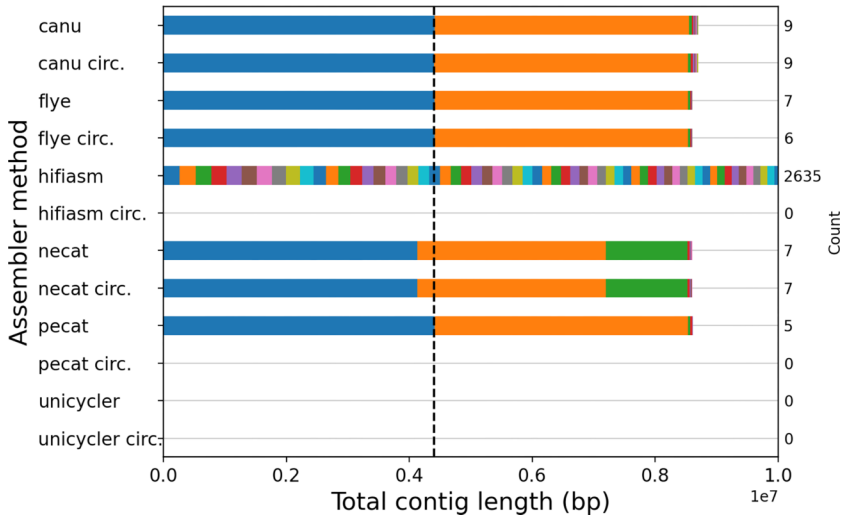


Figure 10. Cyanobacteria assembly results with 366× coverage for the Cyanobacterium test case. Each bar corresponds to the number of contigs obtained (*y*-axis, right) and its size (*x*-axis), depending on the assembler used (*y*-axis, left). The expected outcome is 2 full chromosomes of 2 different cyanobacteria and several plasmids. Most assemblers found the two species and plasmids, represented by different colors on the graph. Hifiasm circ., Pecat circ., and Unicycler failed the task and did not finish. Hifiasm generated >2000 contigs.

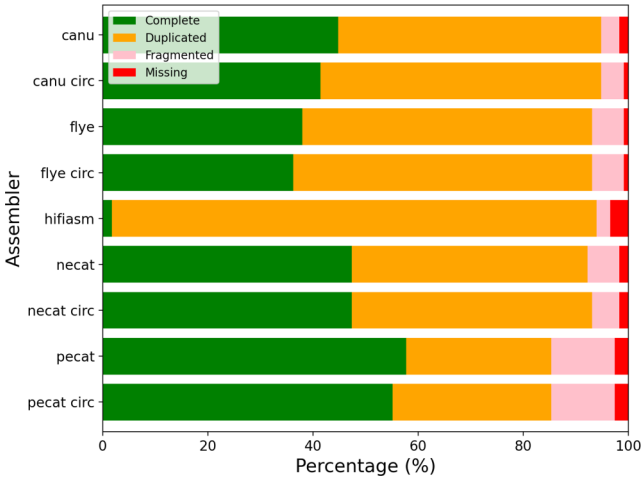


Figure 11. BUSCO scores obtained for the Cyanobacteria case. On the *y*-axis, the assemblers used for this analysis; on the *x*-axis, the Busco score. Due to the contamination with two closely related bacteria, the duplication rate reached 50% in most cases.

As expected, Unicycler, which is optimized for bacterial genomes, struggled with this dataset: it produced 279 contigs totaling 53 Mb, likely reflecting partial phasing of the diploid genome. BUSCO analysis confirmed 20% duplicated genes, consistent with overrepresentation of heterozygous regions.

Overall, this case study highlights that LORA's ecosystem of assemblers extends beyond bacteria. With appropriate tool selection and optional scaffolding, users can assemble small eukaryotic genomes such as *Leishmania* with satisfactory results, even under limited sequencing coverage.

Discussion

Philosophy

LORA was designed to provide a unified, reproducible, and transparent framework for genome assembly using long-read sequencing data. Rather than introducing a new assembler, LORA's main objective is to orchestrate existing best-in-class tools, allowing users to easily compare their performance under consistent and reproducible conditions. The case studies presented in this work demonstrate that assembly outcomes strongly depend on both the input data type and genome complexity. By integrating multiple assemblers, preprocessing steps, and quality assessment modules into a single automated pipeline, LORA enables users to identify the most suitable strategy for their specific datasets.

Comparison with similar pipelines

Several automated genome assembly pipelines have been published in recent years. CulebrONT [46] is a Snakemake-based pipeline designed specifically for ONT data, supporting multiple assemblers (including Flye and Canu), multi-sample processing, and polishing with Medaka and Nanopolish. However, latest version (v4) of CulebrONT does not support PacBio data (CLR or HiFi) and therefore cannot be used for PacBio-only or mixed-technology projects. COLORA [47] is a more recent Snakemake pipeline targeting ONT and PacBio HiFi data, with notable support for Hi-C data integration; however, it relies on a single assembler (Hifiasm) and does not support CCS generation from raw PacBio CLR subreads. MoGAAAP [48] is a Snakemake-based pipeline designed for eukaryotic genome assembly, annotation, and quality assessment, supporting PacBio HiFi and ONT data with Hi-C integration; it focuses on haplotype-resolved assemblies using Hifiasm, Flye, or Verkko. It may be less suited to prokaryotic genomes and does not include read filtering. Among Nextflow-based pipelines, nf-core/bacass supports bacterial genome assembly from short reads, long reads (ONT), and hybrid data with annotation and Kraken2-based contamination detection; however, it is focused on bacterial genomes and ONT data. Another Nextflow pipeline named nf-core/genomeassembler targets long-read (ONT and PacBio HiFi) assembly with polishing and scaffolding, but integrates only two assemblers (Flye and Hifiasm) and does not support PacBio CLR.

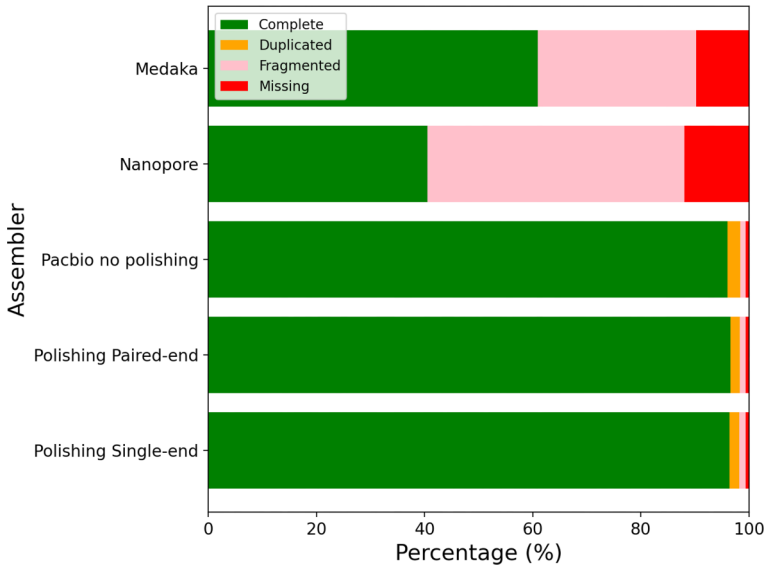


Figure 12. BUSCO scores for the *Bacteroides fragilis* case (mean over 7 samples): raw Nanopore assembly (Flye), Medaka-polished ONT assembly, short-read polishing with Polypolish (paired-end and single-end), and PacBio assembly.

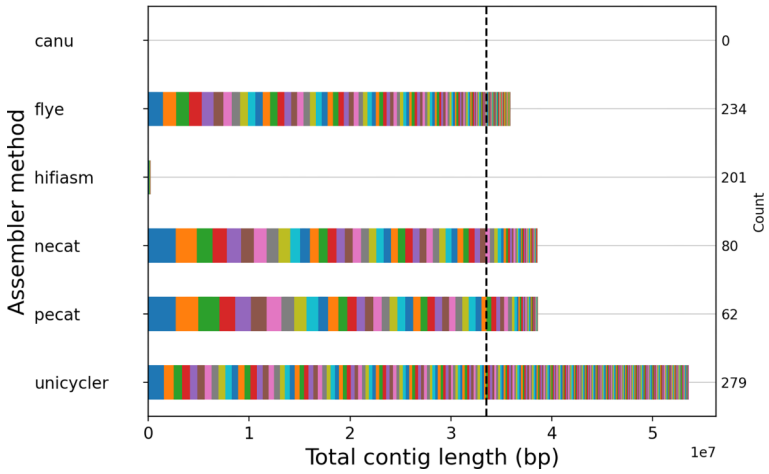


Figure 13. Assembly results for *Leishmania* dataset. Each horizontal bar (divided into colored segments) corresponds to the contigs obtained by different assemblers (*y*-axis, left). The number of contigs is provided (*y*-axis, right) and their total length is indicated on the *x*-axis. The expected outcome is 36 full chromosomes, 1 maxi circle (35 kb), and several mini circles (1 kb) for a total of ∼35 Mb. Canu correction was stopped after 3 days of computation. Total sum of contigs produced by Hifiasm is below 250 000 bases.

LORA complements these efforts rather than replacing them, and the choice between pipelines will depend on the specific use case. Regarding input data, LORA is currently the only pipeline in this comparison that supports on-the-fly CCS generation from raw PacBio CLR subreads, making it applicable to projects where only raw BAM files are available. On the assembler side, LORA integrates six tools—including NECAT and PECAT, which are not offered by the other pipelines and are particularly suited to noisy CLR and ONT reads—though it does not currently support Hi-C data integration, which is available in COLORA and MoGAAAP. For quality assessment, LORA combines BUSCO, CheckM, and QUAST with `sequana-coverage` [30], which provides per-contig depth profiles and CNV detection; this level of coverage diagnostics is not prominently featured in the compared tools. Finally, LORA produces interactive multi-sample HTML reports consolidating assembly statistics, coverage plots, BUSCO and CheckM results, and BLAST-based taxonomic identification in a single document (see Fig. 17 and sequana.github.io/lora-demo); this integrated reporting appears distinctive relative to the pipelines compared here. Taken together, LORA is a complementary option particularly relevant for multi-technology projects, PacBio CLR data, and workflows where integrated quality reporting is a priority.

Reproducibility and FAIR principles

LORA was designed to comply with the principles of reproducible research and the FAIR guidelines (Findable, Accessible, Interoperable, and Reusable) [49]. The workflow is fully version-controlled and documented, ensuring transparency and traceability of all analyses. To guarantee reproducibility across computing environments, LORA uses containerized software dependencies. All non-Python tools, such as minimap2, BUSCO, CheckM, and the included assemblers, are distributed through the Damona project at

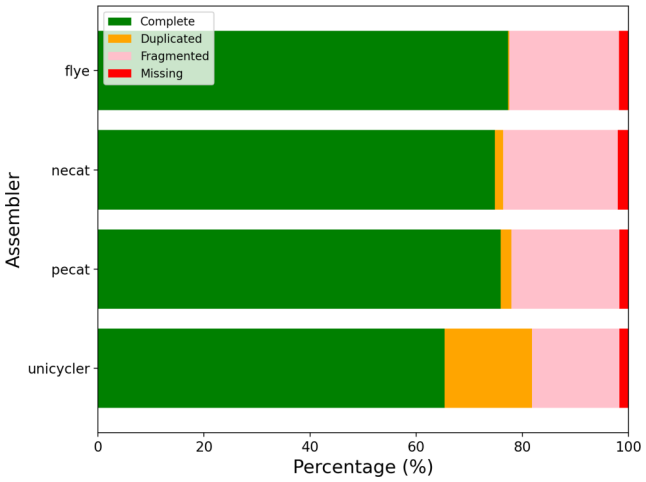


Figure 14. BUSCO scores obtained for the *Leishmania* case. The *y*-axis shows the assemblers used in this analysis, and the *x*-axis indicates the BUSCO score (percentage). Flye, NECAT, and PECAT exhibit similar results, with a large proportion of complete genes and some fragmented ones. Unicycler shows an excess of duplicated genes, while Hifiasm misses most genes due to assembly failure.

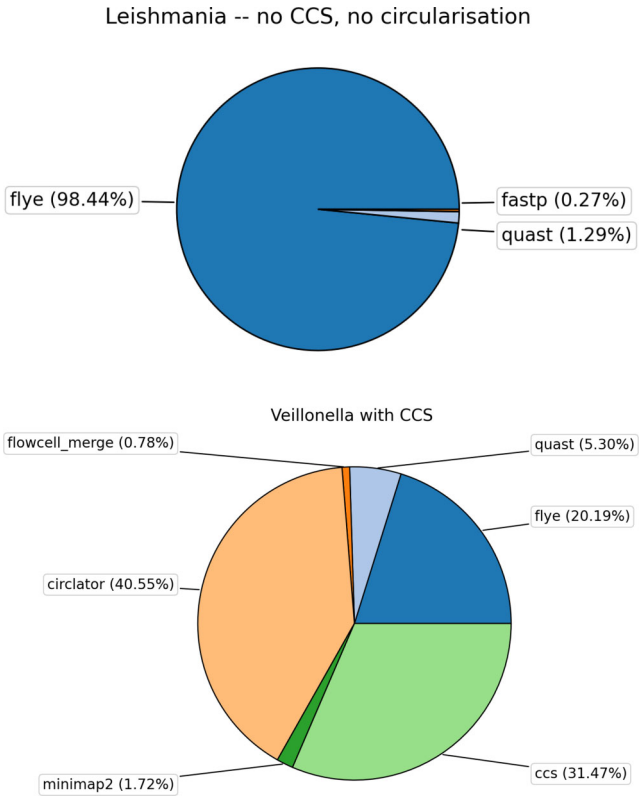


Figure 15. Top pie chart: distribution of CPU time for *Leishmania* data analysis using Flye assembler. Bottom pie chart: distribution of CPU time for the *Veillonella* data analysis from raw PacBio reads, including CCS construction and the circularization step. Tasks requiring <2 min of computation are not shown. When circularization and CCS preprocessing are excluded, the assembly step accounts for the majority of the runtime.

https://damona.readthedocs.io as Apptainer (formerly Singularity) containers [50]. These containers encapsulate all necessary binaries and libraries, eliminating system-level variability. When LORA is executed, the required containers are automatically downloaded and managed by the pipeline, requiring no manual installation or configuration. This approach ensures consistent execution across local machines, HPC clusters, and cloud environments while simplifying deployment for users. By combining transparent configuration, versioned workflows, and containerized dependencies, LORA offers a robust and reproducible framework for long-read genome assembly, fully aligned with FAIR data and software principles. Note that LORA v1.1.0, now available at github.com/sequana/lora, integrates BUSCO v6.0.0 with OrthoDB v12 by default and includes further improvements to the reporting module.

Benchmarking philosophy

LORA is not intended to benchmark assemblers exhaustively but rather to provide the infrastructure that makes fair, reproducible comparison possible. A core value of LORA is precisely that the observed differences in assembly quality are attributable to the assemblers themselves, not to inconsistencies in preprocessing, quality assessment, or execution environment—factors that vary considerably when tools are run in isolation. By controlling for all pipeline steps other than the assembler, LORA turns assembler selection into a first-class scientific question that users can answer rigorously for their own data.

Our case studies illustrate this value concretely. For CLR reads, Flye and Canu consistently produced complete bacterial genomes, while Unicycler and Hifiasm struggled with the same input. For HiFi reads, Flye, Pecat, and Hifiasm yielded highly contiguous assemblies with excellent BUSCO completeness. ONT datasets with potential contamination benefited from LORA's integrated taxonomic analysis, enabling detection of mixed species that would not have been apparent from assembly statistics alone. In diploid eukaryotes, Necat and Pecat captured haplotypic variation, whereas bacterial-genome-oriented assemblers such as Unicycler failed on the same dataset. None of these observations would be reliable without a controlled execution framework—which is precisely what LORA provides. Rather than ranking assemblers globally, LORA empowers users to choose the most appropriate tool based on their specific data type, genome complexity, and computational constraints.

Computational performance and resource management

We quantified the computational demands of each assembler and key workflow steps (Figs 15 and 16). Assembly tasks consistently dominated CPU usage, while circularization and PacBio CCS generation also contributed substantially in bacterial analyses (one third of the run time each). Memory usage varied across assemblers, with Unicycler being the most demanding, and Pecat and Necat showing the best efficiency. Concurrent execution of independent tasks through Snakemake substantially reduces wall-clock time (i.e. real time). For instance, a complete Flye-based assembly of a *Streptococcus* genome, including CCS generation and QC, required ~2 CPU h but <1 h wall-clock time because quality assessment and CCS generation run concurrently as soon as their dependencies are met. This reduction stems from Snakemake scheduling independent jobs in parallel, not from any modification to the threading behavior of the assemblers themselves. The *Bacteroides fragilis* case study illustrates that LORA can ef-

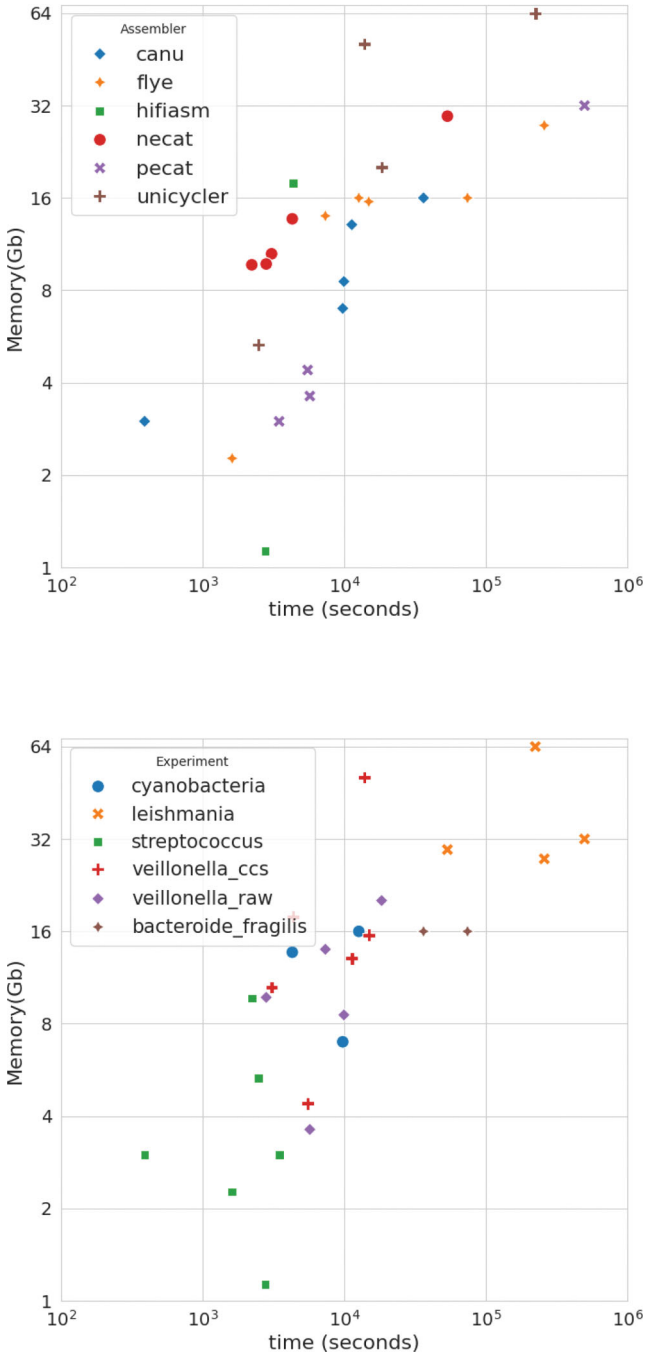


Figure 16. Top panel: CPU time and peak memory usage for all assembler tasks across the different experiments. Bottom panel: CPU time and peak memory usage for all experiment tasks across the different assemblers. In the bottom panel (top right corner), the *Leishmania* assemblies cluster together regardless of the assembler used, whereas on the opposite end of the spectrum lies the SGM case (HiFi data with lower yield). In the top panel, Unicycler shows substantially higher time and memory consumption, while PECAT and NECAT generally exhibit more efficient performance.

ficiently handle both single-sample and multi-sample projects (tested with up to 50 samples), enabling large-scale comparative analyses.

When selecting an assembler, users should balance accuracy and computational efficiency. For small bacterial genomes or constrained environments, PECAT or NECAT provide good completeness with low memory and runtime. Assemblers such as Flye or Canu are more resource-intensive but yield higher contiguity and completeness, particularly for complex bacterial genomes and small eukaryotes. Unicycler may be useful for circular genomes (e.g. missing plasmids), while Hifiasm is ideal for HiFi reads and haplotype-aware assemblies.

Extensibility and modularity

A key advantage of LORA lies in its modular design. The pipeline currently supports six assemblers and Illumina-based polishing with Polypolish or ONT correction (Medaka). Thanks to the Snakemake architecture and containerized modules, new assemblers, polishing tools, or quality control steps can be easily integrated, supporting continuous evolution and community contributions.

Usability

LORA inherits its user interface from the Sequana Pipetools (https://github.com/sequana/sequana_pipetools) framework. It provides a uniform and intuitive command-line experience across all Sequana pipelines. This design ensures that common actions such as launching the pipeline on a local machine or high-performance computing (HPC) cluster, accessing intermediate results, or resuming interrupted runs are performed using standardized commands. Users familiar with other Sequana pipelines can therefore adopt LORA seamlessly without additional learning overhead.

In addition to these shared features, LORA introduces specific options to streamline assembly analyses. For example, the `--mode` option automatically configures workflow parameters for bacterial or eukaryotic genomes. When `--mode` bacteria is selected, the pipeline enables BUSCO and CheckM quality assessments, adjusts coverage analysis accordingly, and triggers Prokka for genome annotation. Similarly, the `--data-type` option standardizes parameters for different sequencing technologies such as Oxford Nanopore or PacBio by harmonizing assembler-specific arguments that are otherwise inconsistent across tools. Concrete examples of typical command-line invocations are provided in Supplementary Listings L1–L5.

Although the initialization command sets all essential parameters, LORA automatically generates a Snakemake-compatible config file populated with the full set of pipeline parameters and their current values. This file can be inspected and freely edited before launching the pipeline, providing fine-grained control without requiring long command lines. This design follows standard Snakemake practice: the YAML configuration file serves as a self-documenting, version-controllable record of all analysis parameters, facilitating reproducibility and sharing.

LORA also automates several tasks to simplify setup. When BUSCO or CheckM analyses are requested, the corresponding reference databases are downloaded on the fly prior to execution, ensuring analyses run smoothly without manual preparation. LORA performs database name validation to detect typographical errors and suggests corrections using a distance-based algorithm.

If a run is interrupted—for instance, due to a job-scheduler timeout or an out-of-memory error on a particular assembler—LORA can be restarted from the same working directory without any modification. Snakemake automatically identifies which rules have already completed successfully (based on the presence of output files and timestamps)

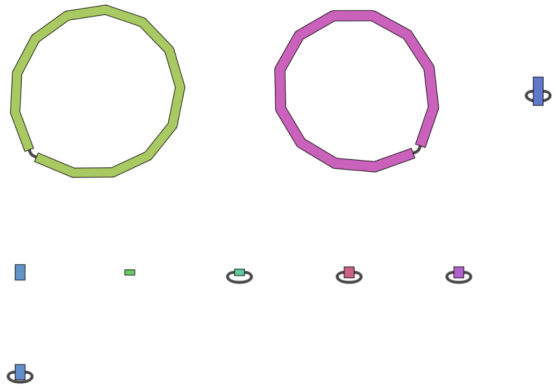
A

Assembly statistics (BUSCO v6.0.0, odb12)

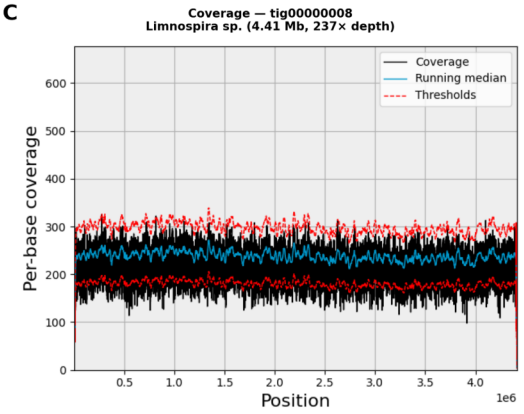
Sample	# contigs	Largest contig (bp)	Total length (bp)	Mapped (%)	Avg. cov.	Complete BUSCOs	Dup. BUSCOs	Frag.	Miss.
sample1	9	4,413,709	8,699,157	100.0	328×	121	69	3	0

B

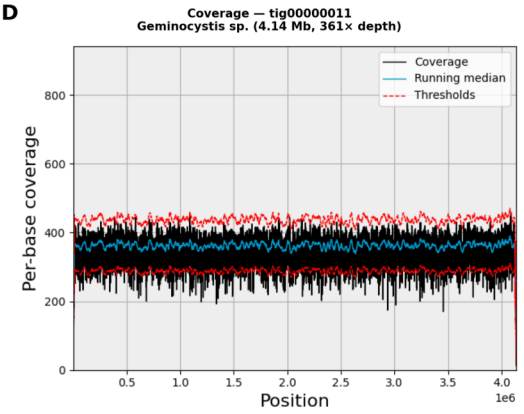
Assembly graph (Flye, Bandage) — two main connected components correspond to the two cyanobacterial chromosomes



C



D



E

BLAST top hits — tig00000008

Identity (%)	Aln. length	E-value	Bitscore	Subject title
88.738	3099	0.0	3784.0	Limnospira sp. BM01 chromosome
88.706	3099	0.0	3779.0	Limnospira fusiformis SAG 85.79 chromosome...
88.641	3099	0.0	3768.0	Arthrospira sp. str. PCC 8005 chromosome, ...
88.545	3099	0.0	3751.0	Arthrospira sp. TJSD092 chromosome, comple...
87.677	3100	0.0	3602.0	Arthrospira platensis YZ genome

F

BLAST top hits — tig00000011

Identity (%)	Aln. length	E-value	Bitscore	Subject title
89.511	9801	0.0	12327.0	Geminocystis sp. NIES-3708 DNA, complete g...
93.058	5128	0.0	7395.0	Geminocystis sp. NIES-3709 DNA, complete g...
87.848	4962	0.0	5770.0	Cyanobacterium aponinum PCC 10605 chromoso...
84.85	4878	0.0	4863.0	Cyanobacterium stanieri PCC 7202, complete...
84.262	4880	0.0	4693.0	Cyanobacterium sp. HL-69 chromosome, compl...

Figure 17. Example HTML report generated by LORA for the Cyanobacteria case study. (**A**) Assembly statistics table: 9 contigs, 8.7 Mb total length, 100% reads mapped, average coverage 328×, and 69 out of 121 complete BUSCO genes duplicated–the first signal of a mixed sample. (**B**) CheckM result confirming 89.6% contamination and 99.2% completeness, consistent with two co-assembled genomes. (**C**) Sequana coverage plot for the largest contig (tig00000008, 4.41 Mb, 237× depth). (**D**) Sequana coverage plot for the largest contig (tig00000011, 4.13 Mb, 362× depth). (**E**) BLAST top hits for tig00000008, identifying the contig as *Limnospira* sp. (**F**) BLAST top hits for tig00000011, identifying the contig as *Geminocystis* sp. Together, these panels illustrate how LORA's integrated report allows users to detect and interpret complex assembly outcomes—here, the co-assembly of two cyanobacterial species—that may not be apparent from any single metric alone.

and resumes execution from the point of failure, avoiding redundant recomputation of earlier steps.

Together, these features make LORA accessible to users with varying levels of bioinformatics expertise, reducing configuration errors and ensuring consistent execution across computational environments.

Reporting

LORA also provides a comprehensive HTML report summarizing key metrics for each sample. The reports include coverage plots, BUSCO and CheckM completeness, assembly statistics, and taxonomic identifications derived from BLAST results. An interactive example report is publicly available at sequana.github.io/lora-demo. Figure 17 illustrates the different sections of such a report using the cyanobacteria case study (Canu assembler, no circularization), which is a particularly compelling demonstration: the assembly statistics already flag an anomaly (57% of complete BUSCO genes are duplicated), CheckM confirms 89.6% contamination, and the per-contig coverage plot reveals a chromosome with uniform depth. The BLAST-based taxonomic identification panel further pinpoints the main contig as *Limnospira* sp. (closely related to *Arthrospira*). This combination of metrics provides an integrated view of assembly quality that would be difficult to obtain by running these tools independently. Although the size of HTML files may grow with large multi-sample projects, this limitation can be mitigated by generating per-sample reports or summary dashboards in future releases. Interactive assembly graphs generated by Flye can also be visualized using Bandage [51], facilitating the exploration of repeat regions and structural complexity.

Conclusion

LORA provides a flexible and reproducible framework for long-read genome assembly, integrating multiple state-of-the-art assemblers and supporting diverse sequencing technologies, including PacBio and Oxford Nanopore. Its modular design allows users to tailor workflows for bacterial or eukaryotic genomes, apply circularization when appropriate, and incorporate short-read polishing to improve assembly accuracy. By supporting haplotype-aware assemblers such as PECAT and Hifiasm, LORA can reconstruct complex diploid genomes while efficiently handling smaller bacterial replicons.

The pipeline's polymorphic nature accommodates both single- and multi-sample experiments, while its containerized implementation ensures full reproducibility across computational environments. Through comprehensive quality assessments, including BUSCO, CheckM, N50, and coverage metrics, users can reliably evaluate assembly completeness, contiguity, and potential contamination. LORA has already been successfully applied in published studies, including *Streptococcus* and *Veillonella* genomes [6, 32], demonstrating its utility for real-world genomics projects.

By combining performance, versatility, and transparency, LORA streamlines the assembly process and provides a robust platform for benchmarking, comparative evaluation of assemblers, and reproducible genomic analyses. Its adherence to FAIR principles and open-source design facilitates reuse, extension, and adoption by the broader research community.

Thanks to its Snakemake-based architecture and containerized modules, LORA remains easily extensible, enabling seamless integration of new assemblers and quality control tools as the field evolves.

Acknowledgements

The authors thank Muriel Gugger (Institut Pasteur, Paris) for sharing the unpublished cyanobacterial dataset used in this study; Christophe Beloin (Institut Pasteur, Paris) for authorizing the use of the Veillonella BAM dataset; and Shaynoor Dramsi (Institut Cochin) for authorizing the use of the *Streptococcus* BAM dataset. The authors also thank the Biomics platform (Institut Pasteur, Paris, France) for sequencing the *Veillonella*, *Streptococcus*, and Cyanobacterial data sets. Graphical Abstract was made with BioRender and is accessible at https://biorender.com/9zygy4z. Figure 1 was made with ChatGPT starting from a draft sketch.

Author contributions: Dimitri Desvillechabrol (Conceptualization [lead], Methodology [lead], Software [equal], Writing – review & editing [equal]), Rania Ouazahrou (Data curation [equal], Formal analysis [lead], Investigation [equal], Writing – review & editing [equal]), Juliana Pipoli da Fonseca (Resources [equal], Validation [supporting], Writing – review & editing [equal]), Gerald F. Späth (Resources [equal], Validation [supporting], Writing – review & editing [equal]), and Thomas Cokelaer (Conceptualization [equal], Data curation [equal], Formal analysis [equal], Project administration [lead], Software [equal], Supervision [lead], Validation [supporting], Writing – original draft [lead])

Supplementary data

Supplementary data is available at NAR Genomics & Bioinformatics online.

Funding

This work was supported by the France Génomique consortium [ANR10-INBS-09-08 and ANR-10-INBS-09-10] and ERC Synergy project DecoLeishRN (grant agreement No. 101071613).

Data availability

All assemblies were performed with LORA v1.0.1, Sequana v0.19.2, Sequana Pipetools v1.0.3, and Snakemake v7.32.4. Note that LORA v1.0.1 ships with BUSCO v5.4.6; however, the BUSCO results presented here were obtained using BUSCO v6.0.0 with OrthoDB v12 (odb12) lineage databases [25], run independently. BUSCO v6.0.0 is now integrated by default in LORA v1.1.0, available at github.com/sequana/lora, so future analyses will not require this separate step. A complete list of all third-party tool versions is provided in Supplementary Table S1. All tools are distributed as Apptainer containers through the Damona project (damona.readthedocs.io). Input sequencing data are provided in the text as Zenodo links or accession numbers.

Figures (except Fig. 1) and tables were generated with the Python notebooks available at https://github.com/cokelaer/paper_LORA and can be executed online (DOI 10.5281/zenodo.17250176).

References

1. Nagarajan N, Pop M. Sequence assembly demystified. *Nat Rev Genet* 2013;14:157–67. https://doi.org/10.1038/nrg3367
2. Zerbino DR, Birney E. Velvet: algorithms for *de novo* short read assembly using de Bruijn graphs. *Genome Res* 2008;18:821–9. https://doi.org/10.1101/gr.074492.107
3. Prjibelski A, Antipov D, Meleshko D *et al.* Using SPAdes *de novo* assembler. *Curr Protoc Bioinform* 2020;70:e102. https://doi.org/10.1002/cpbi.102
4. Mostafa HH. An evolution of Nanopore next-generation sequencing technology: implications for medical microbiology and public health. *J Clin Microbiol* 2024;62:e00246–24. https://doi.org/10.1128/jcm.00246-24
5. Wenger AM, Peluso P, Rowell WJ *et al.* Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nat Biotechnol* 2019;37:1155–62. https://doi.org/10.1038/s41587-019-0217-9
6. Béchon N, Jiménez-Fernández A, Witwinowski J *et al.* Autotransporters drive biofilm formation and autoaggregation in the diderm Firmicute *Veillonella parvula*. *J Bacteriol* 2020;202:e00461–20 . https://doi.org/10.1128/JB.00461-20
7. Périchon B, Cokelaer T, Teh WK *et al.* Colorectal cancer-associated *Streptococcus gallolyticus*: a hidden diversity expose. *J Bacteriol* 2025;207:e00230–25. https://doi.org/10.1128/jb.00230-25
8. Sous C, Frigui W, Pawlik A *et al.* Genomic and phenotypic characterization of *Mycobacterium tuberculosis* closest-related non-tuberculous mycobacteria. *Microbiol Spect* 2024;12:e04126–23. https://doi.org/10.1128/spectrum.04126-23
9. Camacho E, González-de la Fuente S, Rastrojo A *et al.* Complete assembly of the *Leishmania donovani* (HU3 strain) genome and transcriptome annotation. *Sci Rep* 2019;9:6127. https://doi.org/10.1038/s41598-019-42511-4
10. Logsdon GA, Vollger MR, Eichler EE. Long-read human genome sequencing and its applications. *Nat Rev Genet* 2020;21:597–614. https://doi.org/10.1038/s41576-020-0236-x
11. Rhoads A, Au KF. PacBio sequencing and its applications. *Genomics Proteom Bioinform* 2015;13:278–89. https://doi.org/10.1016/j.gpb.2015.08.002
12. Koren S, Walenz BP, Berlin K *et al.* Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res* 2017;27:722–36. https://doi.org/10.1101/gr.215087.116
13. Kolmogorov M, Yuan J, Lin Y *et al.* Assembly of long, error-prone reads using repeat graphs. *Nat Biotechnol* 2019;37:540–6. https://doi.org/10.1038/s41587-019-0072-8
14. Wick RR, Judd LM, Gorrie CL *et al.* Unicycler: resolving bacterial genome assemblies from short and long sequencing reads. *PLoS Comput Biol* 2017;13:e1005595. https://doi.org/10.1371/journal.pcbi.1005595
15. Wick RR, Holt KE. Benchmarking of long-read assemblers for prokaryote whole genome sequencing. *F1000Research* 2020;8:2138. https://doi.org/10.12688/f1000research.21782.3
16. Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. *Genomics* 2010;95:315–27. https://doi.org/10.1016/j.ygeno.2010.03.001
17. Sun J, Li R, Chen C *et al.* Benchmarking Oxford Nanopore read assemblers for high-quality molluscan genomes. *Philos Trans R Soc Lond B Biol Sci* 2021;376:20200160. https://doi.org/10.1098/rstb.2020.0160
18. Koren S, Harhay GP, Smith TPL *et al.* Reducing assembly complexity of microbial genomes with single-molecule sequencing. *Genome Biol* 2013;14:R101. https://doi.org/10.1186/gb-2013-14-9-r101
19. Martin T Ebert P, Marschall T, Read-based phasing and analysis of phased variants with Whatshap. *Methods Mol Biol* 2023;2590:127–139.
20. Weirather JL, de Cesare M, Wang Y *et al.* Comprehensive comparison of Pacific Biosciences and Oxford Nanopore Technologies and their applications to transcriptome analysis. *F1000Research* 2017;6:100. https://doi.org/10.12688/f1000research.10571.2
21. Poplin R, Chang PC, Alexander D *et al.* A universal SNP and small-indel variant caller using deep neural networks. *Nat Biotechnol* 2018;36:983–7. https://doi.org/10.1038/nbt.4235
22. Koren S, Rhie A, Walenz BP *et al.* De novo assembly of haplotype-resolved genomes with trio binning. *Nat Biotechnol* 2018;36:1174–82. https://doi.org/10.1038/nbt.4277
23. Rautiainen M, Nurk S, Walenz BP *et al.* Telomere-to-telomere assembly of diploid chromosomes with Verkko. *Nat Biotechnol* 2023;41:1474–82. https://doi.org/10.1038/s41587-023-01662-6
24. Simão FA, Waterhouse RM, Ioannidis P *et al.* BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics* 2015;31:3210–2. https://doi.org/10.1093/bioinformatics/btv351
25. Tegenfeldt F, Kuznetsov D, Manni M *et al.* OrthoDB and BUSCO update: annotation of orthologs with wider sampling of genomes. *Nucleic Acids Res* 2025;53:D516–22. https://doi.org/10.1093/nar/gkae987
26. Parks DH, Imelfort M, Skennerton CT *et al.* CheckM: assessing the quality of microbial genomes recovered from isolates, single cells, and metagenomes. *Genome Res* 2015;25:1043–55. https://doi.org/10.1101/gr.186072.114
27. Cokelaer T, Cohen-Boulakia S, Lemoine F. Reprohackathons: promoting reproducibility in bioinformatics through training. *Bioinformatics* 2023;39:i11–20. https://doi.org/10.1093/bioinformatics/btad227
28. Köster J, Rahmann S. Snakemake—a scalable bioinformatics workflow engine. *Bioinformatics* 2012;28:2520–2.
29. Cokelaer T, Desvillechabrol D, Legendre R *et al.* 'Sequana': a set of Snakemake NGS pipelines. *J Open Source Softw* 2017;2:352. https://doi.org/10.21105/joss.00352
30. Desvillechabrol D, Bouchier C, Kennedy S *et al.* Sequana coverage: detection and characterization of genomic variations using running median and mixture models. *GigaScience* 2018;7:giy110. https://doi.org/10.1093/gigascience/giy110
31. Hunt M, Silva ND, Otto TD *et al.* Circlator: automated circularization of genome assemblies using long sequencing reads. *Genome Biol* 2015;16:294. https://doi.org/10.1186/s13059-015-0849-0
32. Proutière A, du Merle L, Garcia-Lopez M *et al.* Gallocin A, an atypical two-peptide bacteriocin with intramolecular disulfide bonds required for activity. *Microbiol Spect* 2023;11:e0508522. https://doi.org/10.1128/spectrum.05085-22
33. Sydenham TV, Overballe-Petersen S, Hasman H *et al.* Complete hybrid genome assembly of clinical multidrug-resistant *Bacteroides fragilis* isolates enables comprehensive identification of antimicrobial-resistance genes and plasmids. *Microbial Genom* 2019;5:e000312. https://doi.org/10.1099/mgen.0.000312
34. Almutairi H, Urbaniak MD, Bates MD *et al.* Chromosome-scale genome sequencing, assembly and annotation of six genomes from subfamily *Leishmaniinae*. *Sci Data* 2021;8:234. https://doi.org/10.1038/s41597-021-01017-3
35. Ewels P, Magnusson M, Lundin S *et al.* MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* 2016;32:3047–8. https://doi.org/10.1093/bioinformatics/btw354
36. Caro H, Dollin S, Biton A *et al.* BioConvert: a comprehensive format converter for life sciences. *NAR Genom Bioinform* 2023;5:lqad074. https://doi.org/10.1093/nargab/lqad074
37. Chen S. fastp 1.0: an ultra-fast all-round tool for FASTQ data quality control and preprocessing. *iMeta* 2025;4:e70078. https://doi.org/10.1002/imt2.70078
38. Cheng H, Concepcion GT, Feng X *et al.* Haplotype-resolved *de novo* assembly using phased assembly graphs with hifiasm. *Nat Methods* 2021;18:170–5. https://doi.org/10.1038/s41592-020-01056-5

39. Chen Y, Nie F, Xie SQ *et al.* Efficient assembly of nanopore reads via highly accurate and intact error correction. *Nat Commun* 2021;12:60. https://doi.org/10.1038/s41467-020-20236-7
40. Nie F, Huang N, Zhang J *et al. De novo* diploid genome assembly using long noisy reads. *Nat Commun* 2024;15:2964. https://doi.org/10.1038/s41467-024-47349-7
41. Alonge M, Lebeigle L, Kirsche M *et al.* Automated assembly scaffolding using RagTag elevates a new tomato system for high-throughput genome editing. *Genome Biol* 2022;23:258. https://doi.org/10.1186/s13059-022-02823-7
42. Gurevich A, Saveliev V, Vyahhi N *et al.* QUAST: quality assessment tool for genome assemblies. *Bioinformatics* 2013;29:1072–5. https://doi.org/10.1093/bioinformatics/btt086
43. Seemann T. Prokka: rapid prokaryotic genome annotation. *Bioinformatics* 2014;30:2068–9. https://doi.org/10.1093/bioinformatics/btu153
44. Wick RR, Holt KE. Polypolish: short-read polishing of long-read bacterial genome assemblies. *PLoS Comput Biol* 2022;18:e1009802. https://doi.org/10.1371/journal.pcbi.1009802
45. Oxford Nanopore Technologies. Medaka: sequence correction provided by ONT Research. *GitHub*. 2023. https://github.com/nanoporetech/medaka (3 April 2026, date last accessed).
46. Orjuela J, Comte A, Ravel S *et al.* CulebrONT: a streamlined long reads multi-assembler pipeline for prokaryotic and eukaryotic genomes. *Peer Commun J* 2022;2:e60. https://doi.org/10.24072/pcjournal.153
47. Obinu L, Booth T, De Weerd H *et al.* COLORA: a Snakemake workflow for chromosome-scale genome assembly. *Bioinformatics* 2025;41:btaf175. https://doi.org/10.1093/bioinformatics/btaf175
48. van Workum DJM, Dey KK, Kozik A *et al.* MoGAAAP: a modular genome assembly, annotation and assessment pipeline. *NAR Genom Bioinform* 2026;8:lqag008. https://doi.org/10.1093/nargab/lqag008
49. Wilkinson MD, Dumontier M, Aalbersberg IJ *et al.* The FAIR guiding principles for scientific data management and stewardship. *Sci Data* 2016;3:160018. https://doi.org/10.1038/sdata.2016.18
50. Kurtzer GM, Sochat V, Bauer MW. Singularity: scientific containers for mobility of compute. *PLoS One* 2017;12:1–20. https://doi.org/10.1371/journal.pone.0177459
51. Wick RR, Schultz MB, Zobel J *et al.* Bandage: interactive visualization of *de novo* genome assemblies. *Bioinformatics* 2015;31:3350–2. https://doi.org/10.1093/bioinformatics/btv383

Received: January 21, 2026. **Revised:** April 1, 2026. **Accepted:** April 9, 2026
© The Author(s) 2026. Published by Oxford University Press.
This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (https://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact reprints@oup.com for reprints and translation rights for reprints. All other permissions can be obtained through our RightsLink service via the Permissions link on the article page on our site—for further information please contact journals.permissions@oup.com.