Title

Hybracter: Enabling Scalable, Automated, Complete and Accurate Bacterial Genome Assemblies

Authors

George Bouras[1,2*], Ghais Houtak[1,2], Ryan R. Wick[3], Vijini Mallawaarachchi[4], Michael J. Roach[4,5], Bhavya Papudeshi[4], Lousie M. Judd[3], Anna E. Sheppard[6], Robert A. Edwards[4], Sarah Vreugde[1,2]

Author institutional affiliations

[1] Adelaide Medical School, Faculty of Health and Medical Sciences, The University of Adelaide, Adelaide, Australia.

[2] The Department of Surgery - Otolaryngology Head and Neck Surgery, University of Adelaide and the Basil Hetzel Institute for Translational Health Research, Central Adelaide Local Health Network, South Australia, Australia.

[3] Department of Microbiology and Immunology, University of Melbourne at the Peter Doherty Institute for Infection and Immunity, Melbourne, Australia.

[4] Flinders Accelerator for Microbiome Exploration, College of Science and Engineering, Flinders University, Adelaide, Australia.

[5] Adelaide Centre for Epigenetics and South Australian Immunogenomics Cancer Institute, The University of Adelaide, Adelaide, Australia.

[6] School of Biological Sciences, The University of Adelaide, Adelaide, Australia.

[*] Correspondence: george.bouras@adelaide.edu.au;

Abstract

Improvements in the accuracy and availability of long-read sequencing mean that complete bacterial genomes are now routinely reconstructed using hybrid (i.e. short- and long-reads) assembly approaches. Complete genomes allow a deeper understanding of bacterial evolution and genomic variation beyond single nucleotide variants (SNVs). They are also crucial for identifying plasmids, which often carry medically significant antimicrobial resistance (AMR) genes. However, small plasmids are often missed or misassembled by long-read assembly algorithms. Here, we present Hybracter which allows for the fast, automatic, and scalable recovery of near-perfect complete bacterial genomes using a long-read first assembly approach. Hybracter can be run either as a hybrid assembler or as a long-read only assembler. We compared Hybracter to existing automated hybrid and long-read only assembly tools using a diverse panel of samples of varying levels of long-read accuracy with manually curated ground truth reference genomes. We demonstrate that Hybracter as a hybrid assembler is more accurate and faster than the existing gold standard automated hybrid assembler Unicycler. We also show that Hybracter with long-reads only is the most accurate long-read only assembler and is comparable to hybrid methods in accurately recovering small plasmids.

Data Summary

1. Hybracter is developed using Python and Snakemake as a command-line software tool for Linux and MacOS systems.

46 2. Hybracter is freely available under an MIT License on GitHub
47 (https://github.com/gbouras13/hybracter) and the documentation is available
48 at Read the Docs (https://hybracter.readthedocs.io/en/latest/).
49 3. Hybracter is available to install via PyPI (https://pypi.org/project/hybracter/)
50 and Bioconda (https://anaconda.org/bioconda/hybracter). A Docker/Singularity
51 container is also available at https://quay.io/repository/gbouras13/hybracter.
52 4. All code used to benchmark Hybracter, including the reference genomes, is
53 publicly available on GitHub
54 (https://github.com/gbouras13/hybracter_benchmarking) with released DOI
55 (https://zenodo.org/doi/10.5281/zenodo.10910108) available at Zenodo.
56 5. The subsampled FASTQ files used for benchmarking are publicly available at
57 Zenodo with DOI (https://doi.org/10.5281/zenodo.10906937).
58 6. All super accuracy simplex ATCC FASTQ reads sequenced as a part of this study
59 can be found under BioProject PRJNA1042815.
60 7. All *Hall* et al. fast accuracy simplex and super accuracy duplex ATCC FASTQ read
61 files (prior to subsampling) can be found in the SRA under BioProject
62 PRJNA1087001.
63 8. All raw *Lermaniaux* et al. FASTQ read files and genomes (prior to subsampling)
64 can be found in the SRA under BioProject PRJNA1020811.
65 9. All *Staphylococcus aureus* JKD6159 FASTQ read files and genomes can be found
66 under BioProject PRJNA50759.
67 10. All *Mycobacterium tuberculosis* H37R2 FASTQ read files and genomes can be
68 found under BioProject PRJNA836783.
69 11. The complete list of BioSample accession numbers for each benchmarked sample
70 can be found in Supplementary Table 1.

12. The benchmarking assembly output files are publicly available on Zenodo with DOI (https://doi.org/10.5281/zenodo.10906937).
13. All Pypolca benchmarking outputs and code are publicly available on Zenodo with DOI (https://zenodo.org/doi/10.5281/zenodo.10072192).

Impact Statement

Complete bacterial genome assembly using hybrid sequencing is a routine and vital part of bacterial genomics, especially for identification of mobile genetic elements and plasmids. As sequencing becomes cheaper, easier to access and more accurate, automated assembly methods are crucial. With Hybracter, we present a new long-read first automated assembly tool that is faster and more accurate than the widely-used Unicycler. Hybracter can be used both as a hybrid assembler and with long-reads only. Additionally, it solves the problems of long-read assemblers struggling with small plasmids, with plasmid recovery from long-reads only performing on par with hybrid methods. Hybracter can natively exploit the parallelisation of high-performance computing (HPC) clusters and cloud-based environments, enabling users to assemble hundreds or thousands of genomes with one line of code. Hybracter is available freely as source code on GitHub, via Bioconda or PyPi.

Introduction

Reconstructing complete bacterial genomes using *de novo* assembly methods had been considered too costly and time-consuming to be widely recommended in most cases, even as recently as 2015 [1]. This was due to the reliance on short-read sequencing technologies, which does not allow for reconstructing regions with repeats and extremely high GC content [2]. However, since then, advances in long-read sequencing technologies have allowed for the automatic construction of complete genomes using hybrid assembly approaches. Originally, this involved starting with a short-read assembly followed by scaffolding the repetitive and difficult to resolve regions with long-reads [3,4]. This approach was implemented in the command-line tool Unicycler, which remains the most popular tool for generating complete bacterial genome assemblies [5]. As long-read sequencing has improved in accuracy and availability, with the latest Oxford Nanopore Technologies reads recently reaching Q20 (99%+) median accuracy, a long-read first assembly approach supplemented by short-read polishing has recently been favoured for recovering accurate complete genomes. Long-read-first approaches provide greater accuracy and contiguity than short-read-first approaches in difficult regions [6–11]. The current gold standard manual assembly tool Trycycler even allows for the potential recovery of perfect genome assemblies [7]. However, Trycycler requires significant microbial bioinformatics expertise and involves manual decision making, creating a significant barrier to useability, scalability and automation [12].

Several tools exist that generate automated long-read first genome assemblies, such as MicroPIPE [13], ASA3P [14], Bactopia [15] and Dragonflye [16]. However, these tools do not consider factors such as genome reorientation [17] and recent polishing best-practices [18], and often contain the assembly workflow as a sub-module within a more expansive end-to-end

114 pipeline. Additionally, none of the existing tools consider the targeted recovery of plasmids.
115 As long-read assemblers struggle particularly with small plasmids, this leads to incorrectly
116 recovered or missing plasmids in bacterial assemblies [19].
117
118 We introduce Hybracter, a new command-line tool for automated near-perfect long-read-first
119 complete bacterial genome assembly. It implements a comprehensive and flexible workflow
120 allowing for long-read assembly polished with long and short-reads (with subcommand
121 'hybracter hybrid' for one or more samples and subcommand 'hybracter hybrid-single' for a
122 single sample) or long-read only assembly polished with long-reads (with subcommand
123 'hybracter long' for one or more samples and subcommand 'hybracter long-single' for a
124 single sample) (Table 1). For ease of use and familiarity, Hybracter has been designed with a
125 command-line interface containing parameters similar to Unicycler. Additionally, thanks to
126 its Snakemake [20] and Snaketool [21] implementation, Hybracter seamlessly scales from a single
127 isolate to hundreds or thousands of genomes with high computational efficiency and supports
128 deployment on HPC clusters and cloud-based environments.
129

130 # Methods

131 ## Assembly Workflow

132

133 Hybracter implements a long-read-first automated assembly workflow based on current best
134 practices [12]. The main subcommands available in Hybracter can be found in Table 1 and the
135 workflow is outlined in Figure 1. Hybracter begins with long-reads for all subcommands, and
136 uses short-reads for polishing for 'Hybracter hybrid' and 'Hybracter hybrid-single'
137 subcommands.

138

139 First, long-read input FASTQs are input long read sets are filtered and subsampled to a depth
140 of 100x with Filtlong[22], which prioritises the longest and highest quality reads, outperforming
141 random subsampling (See Supplementary Table 11). The reads also have adapters trimmed
142 using Porechop_ABI[23], with optional contaminant removal against a host genome using
143 modules from Trimnami (e.g. if the bacteria has been isolated from a host) [24]. Quality control
144 of short-read input FASTQs is performed with fastp [25] (Fig 1A). The estimated depth of the
145 short-reads is determined using Seqkit[26].

146

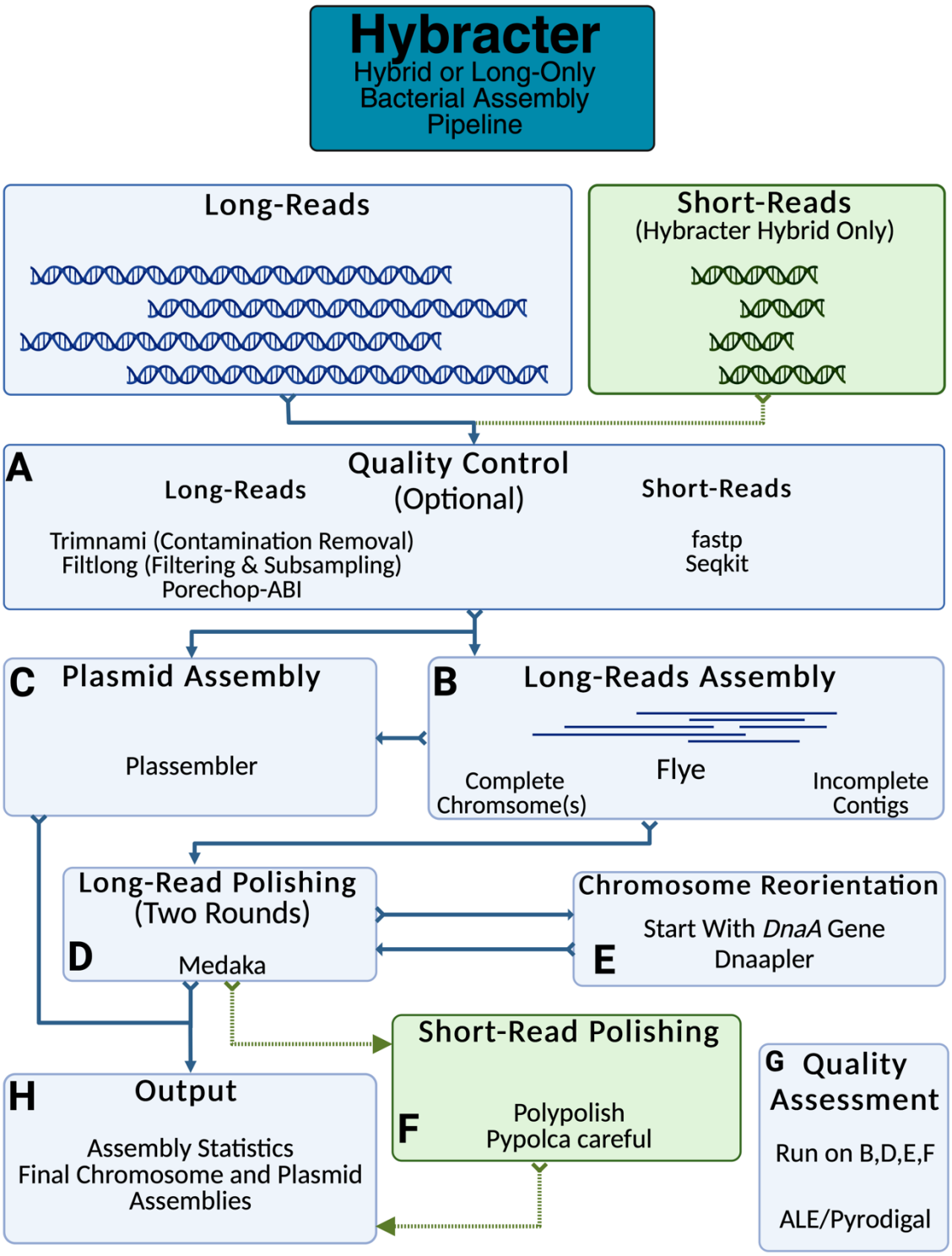
147 Long-reads are then assembled with Flye [27]. If at least 1 contig is recovered above the cut-off
148 '-c' chromosome length specified by the user for the sample, that sample will be denoted as
149 'complete'. All such contigs will then be marked as chromosomes and kept for downstream
150 polishing and reorientation if marked as circular by Flye. If zero contigs are above the cut-off
151 chromosome length, the assembly will be denoted as 'incomplete', and all contigs will be
152 kept for downstream polishing (Fig 1B).

153

154 For all complete samples, targeted plasmid assembly is then conducted using Plassembler [28]
155 (Fig 1C). All samples (i.e. complete and incomplete) are then polished once with Medaka[29],
156 which can be turned off using '--no_medaka' (Fig 1D). It is recommended to turn off Medaka
157 using '--no_medaka' for highly accurate Q20+ read sets where Medaka has been shown to
158 introduce false positive changes[11]. For all complete samples only, chromosome(s) marked as
159 circular by Flye will then be reoriented to begin with the dnaA chromosomal replication
160 initiator gene using Dnaapler[30]. These reoriented chromosomes are then polished for a second
161 time with Medaka to ensure the sequence around the original chromosome breakpoint is
162 polished.
163
164 If the user has provided short-reads with Hybracter hybrid, all samples' assemblies (complete
165 and incomplete) are then polished with Polypolish [18] followed by Pypolca [31,32] (Fig 1F). The
166 exact parameters depend on the depth of short-read sequencing[31]. If the estimated short-read
167 coverage is below 5x, only Polypolish with '--careful' is run, as Pypolca can rarely introduce
168 false positive errors at low depths. If the estimated short-read coverage is between 5-25x,
169 Polypolish with --careful parameter is run followed by Pypolca with --careful parameter.
170 Above 25x coverage, Polypolish with default parameters followed by Pypolca with --careful
171 is run. This is because Pypolca --careful has been shown to be the best polisher at depths
172 above 5x, and because Polypolish is able to fix potential errors in repeats Pypolca may miss.
173 By default, the last short-read polishing round is chosen as the final assembly. Alternatively,
174 users can choose the highest scoring polishing round according to the reference-free ALE [33]
175 score.
176
177 If only long-reads are available (Hybracter long), the mean coding sequence (CDS) length is
178 calculated for each assembly using Pyrodigal [34,35], with larger mean CDS lengths indicating a

179 better quality assembly. The polishing round with the highest mean CDS length is chosen as
180 the final assembly (Fig 1G).
181
182 For each sample, a final output assembly FASTA file is created, along with per contig and
183 overall summary statistic TSV files, as well as separate chromosome and plasmid FASTA
184 files for samples denoted as complete (Fig 1H). An overall 'hybracter_summary.tsv' file is
185 also generated, which summarises outputs for all samples. All main output files are explained
186 in more detail in Table 2. All the main outputs can be found in the 'FINAL_OUTPUT'
187 subdirectory, while all other intermediate output files are available in other subdirectories for
188 users who would like extra information about their assemblies, including all assembly
189 assessments, comparisons of all changes introduced by polishing, and Flye and Plassembler
190 output summaries. A full list of these supplementary outputs can be found in Hybracter's
191 Documentation (https://hybracter.readthedocs.io/en/latest/output/).
192

193 Figure 1: Outline of the Hybracter workflow.



194

195 Table 1. Summary of the 4 Primary Hybracter Commands

Command	Input	Number of Samples	Description	Workflow Elements Included by Default (From Figure 1)
hybracter hybrid	5 column csv sample sheet specified with '--input' containing: • sample name • long-read FASTQ path, • estimated chromosome length • R1 short-read FASTQ path • R2 short-read FASTQ path	1+	Long-read first assembly followed by long then short-read polishing for multiple isolates. Snakemake implementation ensures efficient use of available resources	A, B, C, D, E, F, G, H
hybracter hybrid-single	• sample name (-s) • long-read FASTQ path (-l) • estimated chromosome length (-c) • R1 short-read FASTQ path (-1) • R2 short-read FASTQ path (-2)	1	Long-read first assembly followed by long then short-read polishing for a single isolate. Similar command line interface to Unicycler	A, B, C, D, E, F, G, H
hybracter long	3 column csv sample sheet specified with '--input' containing: • sample name • long-read FASTQ path, • estimated chromosome length	1+	Long-read first assembly followed by long-read polishing for multiple isolates. Snakemake implementation ensures efficient use of available resources	A (no fastp), B, C, D, E, G, H
hybracter long-single	• sample name (-s) • long-read FASTQ path (-l) • estimated chromosome length (-c)	1	Long-read first assembly followed by long-read polishing on a single isolate.	A (no fastp), B, C, D, E, G, H

196

197

198 Table 2. Description of the Primary Hybracter Output Files

Output File	Description
{sample}_final.fasta	Final assembly FASTA file for the sample. Contains all chromosome(s) and plasmids for complete isolates and all contigs for incomplete isolates.
{sample}_chromosome.fasta	Final assembly FASTA file for the chromosomes(s) in a complete sample.
{sample}_plasmid.fasta	Final assembly FASTA file for the plasmids in a complete sample.
hybracter_summary.tsv	A TSV file combining the {sample}_summary.tsv files for all samples.
{sample}_summary.tsv	A TSV file containing columns denoting for the sample: • Assembly completeness • Total assembly length • Number of contigs assembled • The polishing round deemed to be most accurate and selected as the final assembly • The length of the longest contig • The estimated coverage of the longest contig • The number of circular plasmids recovered by Plassembler
{sample}_per_contig_stats.tsv	A TSV file containing columns denoting for the sample: • Contig name • Contig Type (chromosome or plasmid) (complete samples only) • Contig length • Contig GC% • Contig circularity (complete samples only)

199

200 Tool Selection

201

202 Tools were selected for inclusion in Hybracter either based on benchmarking from the
203 literature, or they were specifically developed for inclusion in Hybracter. Flye [27] was chosen
204 as the long-read assembler because it is more accurate for bacterial genome assembly than
205 other long-read assemblers with comparable runtimes, such as Raven [36], Redbean [37] and
206 Miniasm [38], while being dramatically faster than the comparably accurate Canu [6,39]. Medaka
207 [29] was chosen as the long-read polisher because of its ability to improve assembly continuity
208 in addition to accuracy [12,40]. The benchmarking results of this study also emphasise that it is
209 particularly good at fixing insertion and deletion (InDel) errors, which cause problematic
210 frameshifts and frequently lead to fractured or truncated gene predictions. However, it should
211 be re-iterated that for modern Q20+ datasets, Medaka may introduce errors[11] and should not
212 be used (using --no_medaka with Hybracter). Polypolish and Pypolca in various
213 combinations depending on short-read depth were selected as short-read polishers, as these
214 have been shown to achieve the highest performance with the lowest chance of introducing
215 errors when used in combination [31].

216

217 We developed three standalone programs included in Hybracter. These are Dnaapler[30],
218 Plassembler[28] and Pypolca[31]. Dnaapler was developed to ensure the chromosome(s) identified
219 by Hybracter are reoriented to consistently begin with the dnaA chromosomal replication
220 initiator gene. Full implementation details can be found in the manuscript, with expanded
221 functionality beyond this use case[30]. Plassembler was developed to improve the runtime and
222 accuracy when assembling plasmids in bacterial isolates. Full implementation details can be
223 found in the manuscript for hybrid mode [28]. Hybracter long utilises Plassembler containing a
224 post-publication improvement for long-reads only ('Plassembler long') released in v1.3.

225 Plassembler long assembles plasmids from only long-reads by treating long-reads as both
226 short-reads and long-reads. Plassembler long does this by utilising Unicycler in its pipeline to
227 create a de Bruijn graph-based assembly, treating the long-reads as unpaired single-end reads,
228 which are then scaffolded with the same long-read set.
229
230 The third tool is Pypolca[31,32]. Pypolca is a Python re-implementation of the POLCA short-
231 read genome polisher, originally created specifically for inclusion in Hybracter and with an
232 almost identical output format and performance. Compared to POLCA, Pypolca features
233 improved useability with a simplified command line interface, allows the user to specify an
234 output directory and introduces a '--careful' parameter. The performance of Pypolca, and
235 particularly Pypolca with the --careful parameter, are described in the manuscript[31].

236 Benchmarking

237 To compare Hybracter's functionality and performance, we benchmarked its performance
238 against other software tools. We focused on the most popular state-of-the-art assembly tools
239 for automated hybrid and long only bacterial genome assemblies. All code to replicate these
240 analyses can be found at the repository
241 (https://github.com/gbouras13/hybracter_benchmarking). All programs and dependency
242 versions used for benchmarking can be found in Supplementary Table 4. For the hybrid tools,
243 we chose Unicycler and Dragonflye with both long-read and short-read polishing (denoted
244 'Dragonflye hybrid'). Dragonflye was chosen as it is a popular long-read first assembly
245 pipeline [16]. Both tools were run using default parameters. By default, Dragonflye conducts a
246 long-read assembly with Flye that is polished by Racon[41] followed by Polypolish. For the
247 long-read only tool, we chose Dragonflye with long-read Racon based polishing only
248 (denoted 'Dragonflye long').

249

250 We used 30 samples for benchmarking, representing genomes from a variety of Gram-
251 negative and Gram-positive bacteria. We chose these samples as they have real hybrid read
252 sets in combination with manually curated genome assemblies produced using either
253 Trycycler or Bact-builder[42]—a consensus-building pipeline based on Trycycler. These
254 samples came from 5 different studies below. We used the published genomes from these
255 studies as representatives of the 'ground truth' for these samples. Where read coverage
256 exceeded 100x samples were subsampled to approximately 100x coverage of the approximate
257 genome size with Rasusa v0.7.0[43] , as this better reflects more realistic read depth of real life
258 isolate sequencing. Nanoq v0.10.0[44] was used to generate quality control statistics for the
259 subsampled long-read sets. Four isolates did not have 100x long-read coverage — the entire
260 long-read set was used instead. A full summary table of the read lengths, quality, Nanopore
261 kit and base-calling models used in these studies can be found in Supplementary Table 2.
262 Hybracter v0.7.0 was used to conduct benchmarking. Medaka long-read polishing was used
263 for all samples except the 5 ATCC super-accuracy model basecalled duplex read samples,
264 where '--no_medaka' was used.
265

266 These samples contained varying levels of long-read quality (reflecting improvements in
267 Oxford Nanopore Technologies long-read technology), with the median Q score of long-read
268 sets ranging from 10.6 to 26.8. The five studies are:

269 1. Five ATCC strain isolates (ATCC-10708 *Salmonella enterica*, ATCC-17802 *Vibrio*
270 *paragaemolyticus*, ATCC-25922 *Escherichia coli*, ATCC-33560 *Campylobacter*
271 *jejuni* and ATCC-BAA-679 *Listeria monocytogenes*) with R10 chemistry super-
272 accuracy model basecalled simplex long-reads made available as a part of this study.

2. The same 5 ATCC isolates with R10 chemistry fast model basecalled long-reads, and R10 chemistry super-accuracy model basecalled duplex long-reads from *Hall* et al.[45]
3. Twelve diverse carbapenemase-producing Gram-negative bacteria from *Lerminiaux* et al.[9]
4. *Staphylococcus aureus* JKD6159 sequenced with both R9 and R10 chemistry long-read sets from *Wick* et al.[46]
5. *Mycobacterium tuberculosis* HR37v from *Chitale* et al.[42]

The full details for each individual isolate used can be found in Supplementary Tables 1 and 2.

Chromosome Accuracy

The assembly accuracy of the chromosomes recovered by each benchmarked tool was compared using Dnadiff v1.3 packaged with MUMmer v3.23[47]. Comparisons were performed on the largest assembled contig (denoted as the chromosome) by each method, other than for ATCC-17802 *Vibrio parahaemolyticus*, where the two largest contigs were chosen as it has two chromosomes.

Plasmid Recovery Performance and Accuracy

Plasmid recovery performance for each tool was compared using the following methodology. Summary statistics are presented in Table 4. See Supplementary Table 7 for a full sample-by-sample analysis. All samples were analysed using the 4-step approach outlined below using summary length and GC% statistics for all contigs and the output of Dnadiff v1.3

296 comparisons generated for each sample and tool combination against the reference genome
297 plasmids:
298
299 1. The number of circularised plasmid contigs recovered for each isolate was compared
300 to the reference genome. If the tool recovered a circularised contig homologous to that
301 in the reference, it was denoted as completely recovered. Specifically, a contig was
302 denoted as completely recovered if it had a genome length within 250bp of the
303 reference plasmid, a GC% within 0.1% of the reference plasmid and whether the
304 Total Query Bases covered was within 250bp of the Total Reference Bases from
305 Dnadiff. For Dragonflye assemblies, some plasmids were duplicated or multiplicated
306 due to known issues with the long-read first assembly approach for small plasmids
307 [6,19,48]. Any circularised contigs that were multiplicated compared to the reference
308 plasmid were therefore denoted as misassembled.
309 2. For additional circularised contigs not found in the reference recovered, these were
310 tested for homology with NCBI nt database using the web version of blastn[48]. If there
311 was a hit to a plasmid, the Plassembler output within Hybracter was checked for
312 whether the contig had a Mash hit (i.e. a Mash distance of 0.2 or lower) to plasmids in
313 the PLSDB[49]. If there was a hit, the contig was denoted as an additional recovered
314 plasmid. There were 2 in total (see Supplementary Table 7 and supplementary data).
315 3. Plasmids with contigs that were either not circularised but homologous to a reference
316 plasmid, or circularised but incomplete (failing the genome length and Dnadiff criteria
317 in 1.) were denoted as partially recovered or misassembled.
318 4. Reference plasmids without any homologous contigs in the assembly were denoted as
319 missed.
320

321 Additional non-circular contigs that had no homology with reference plasmids and were not
322 identified as plasmids in step 2 were analysed on a contig-by-contig basis and denoted as
323 additional non-plasmid contigs (see Supplementary Table 7 for contig-by-contig analysis
324 details).
325

326 ## Runtime Performance Comparison

327

328 To compare the performance of Hybracter, we compared wall-clock runtime consumption on
329 a machine with an Intel® Core™ i9-13900 CPU @ 5.60 GHz on a machine running Ubuntu
330 20.04.6 LTS with a total of 32 available threads (24 total cores). We ran all tools with 8 and
331 16 threads and with 32 GB of memory to provide runtime metrics comparable to commonly
332 available consumer hardware. Hybracter hybrid and long were run with 'hybracter hybrid-
333 single' and 'hybracter long-single' for each isolate to generate a comparable per sample
334 runtime for comparison with the other tools. The summary results are available in Table 5
335 and the detailed results for each specific tool and thread combination are found in
336 Supplementary Table 8.
337

338 ## Depth Analysis

339 To assess the effect of long-read depth on assembly accuracy, we chose *Lerminiaux* Isolate B
340 (*Enterobacter cloacae*) and subsampled the long-read depth at each interval of 5x from 10x
341 to 100x estimated genome size. All 5 tools were run on these read sets. Where a complete
342 chromosome was assembled, Dnadiff (as described above) was used to compare the
343 chromosome assembly to the reference.

344 ## Sequencing

345

346 DNA extraction was performed with the DNeasy Blood and Tissue kit (Qiagen). Illumina
347 library preparation was performed using Illumina DNA prep (Illumina Inc.) according to the
348 manufacturer's instructions. Short-read whole genome sequencing was performed an Illumina
349 MiSeq with a 250bp PE kit. Oxford Nanopore Technologies library preparation ligation
350 sequencing library was prepared using the ONT SQK-NBD114-96 kit and the resultant
351 library was sequenced using an R10.4.1 MinION flow cell (FLO-MIN114) on a MinION
352 Mk1b device. Data was base-called with Super-Accuracy Basecalling (SUP) using the
353 basecaller model dna_r10.4.1_e8.2_sup@v3.5.1.
354

355 ## Pypolca Benchmarking

356

357 Pypolca v0.2.0 was benchmarked against POLCA (in MaSuRCA v4.1.0)[32] using 18 isolates
358 described above. These were all 12 *Lerminiaux* et al. isolates, the R10 JKD6159 isolate [46] and
359 the 5 ATCC samples we sequenced as a part of this study. Benchmarking was conducted on
360 an Intel® Core™ i7-10700K CPU @ 3.80 GHz on a machine running Ubuntu 20.04.6 LTS.
361 All short read FASTQs used for benchmarking are identical to those used to benchmark
362 Hybracter. The assemblies used for polishing were intermediate chromosome assemblies
363 from Flye v2.9.2[50] generated within Hybracter. The outputs from Pypolca and POLCA were
364 compared using Dnadiff v1.3 packaged with MUMmer v3.23[47] Overall, Pypolca and POLCA
365 yielded extremely similar results. 16/18 assemblies were identical. ATCC 33560 had 2 Single
366 Nucleotide Polymorphisms (SNPs) between Pypolca and POLCA and *Lerminiaux* Isolate I
367 also had 2 SNPs.

368 # Results

369 ## Chromosome Accuracy Performance

370

371 All tools recovered complete circular contigs for each chromosome. SNVs, small InDels
372 (under 60 bps), and large InDels (over 60 bps) were compared as a measure of assembly
373 accuracy. To account for differences in genomic size between isolates, SNVs and small InDel
374 counts were normalised by genome length.

375

376 The summary results are presented in Table 3 and visualised in Figure 2. The detailed results
377 for each tool and sample are presented in Supplementary Table 5. Of the hybrid tools,
378 Dragonflye hybrid and Hybracter hybrid produced the fewest SNVs (both with median 0)
379 followed by Unicycler (median 34). Hybracter hybrid produced the fewest InDels (median 0),
380 followed by Dragonflye hybrid (median 2.5) and Unicycler (median 11). Hybracter hybrid
381 also produced the fewest InDels plus SNVs (median 1), followed by Dragonflye hybrid
382 (median 4.5) and Unicycler (median 57.5).

383

384 Additionally, Hybracter hybrid showed superior performance in terms of large InDels, with a
385 median of 0 and a total of 9 large InDels across the 30 samples, compared to 2 and 70 for
386 Dragonflye hybrid, and 1 and 87 for Unicycler.

387

388 Overall, Hybracter hybrid produced the most accurate chromosome assemblies. For twelve
389 isolates, Hybracter assembled a perfect chromosome (*Lerminiaux* et al.[9] Isolates A, B, C, D,
390 G, H, I, J, L, *S. aureus* JKD6159 with R10 chemistry and *L. monocytogenes* ATCC BAA-679
391 with simplex and duplex super-accuracy model basecalled reads).

392

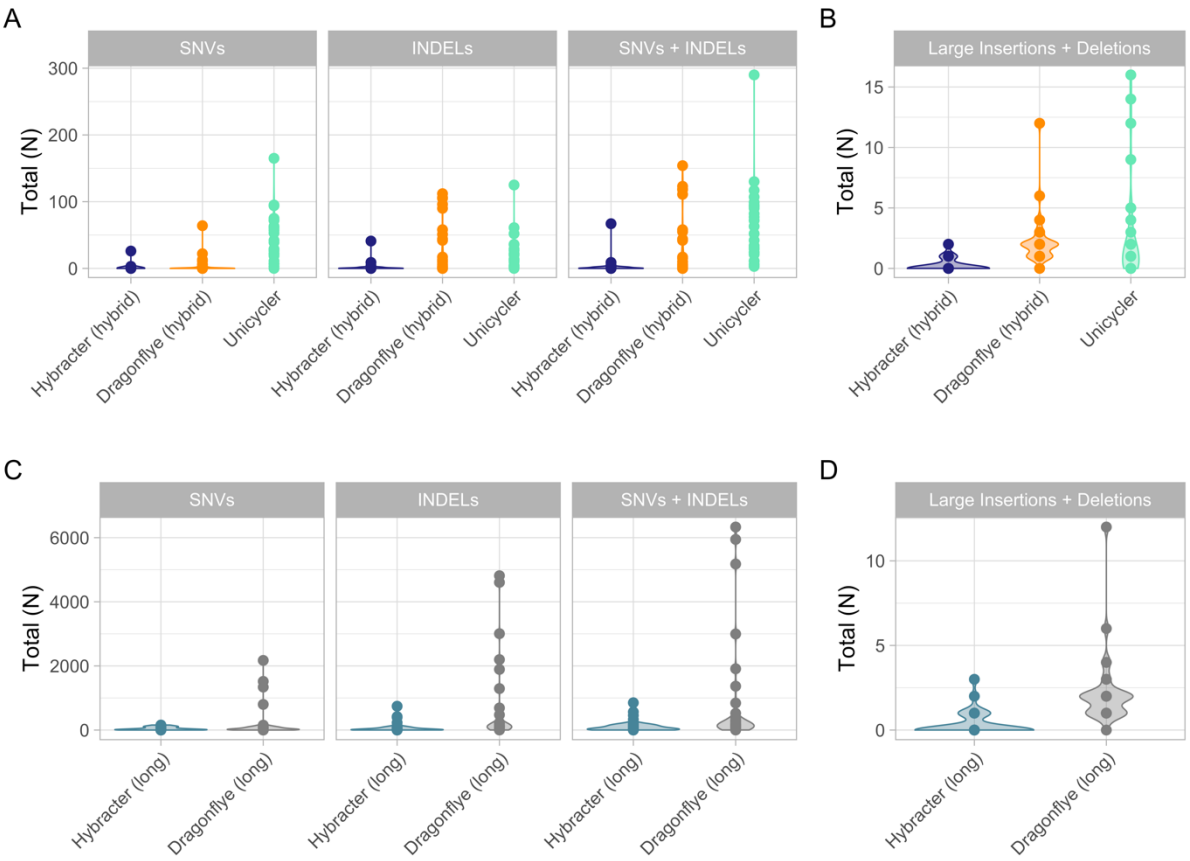
393 Hybracter hybrid also produced several near-perfect assemblies (defined as <10 total SNVs

394 plus InDels with no large insertions or deletions), including on some lower quality fast model

395 basecalled reads (Supplementary Table 5).

396

397 Figure 2: Comparison of the counts of single nucleotide variants (SNVs) and small (<60bp)
398 insertions and deletions (InDels) (A) and the total number of large (>60bp) InDels (B) for
399 the hybrid tools benchmarked (Hybracter hybrid in dark blue, Dragonflye hybrid in orange
400 and Unicycler in green). The counts of SNVs and small InDels (C) and the total number of
401 large InDels (D) for the long tools benchmarked (Hybracter long in light blue, Dragonflye
402 long in grey) are also shown. All data presented is from the benchmarking output run
403 with 8 threads.



404
405

406 Table 3. Small (<60bp) InDels, SNVs and large (>60bp) InDels of Chromosomes

407 Assemblies for all Benchmarked Isolates.

408

Tool	Type	Small InDels	SNVs	Small InDels + SNVs	Large InDels
Hybracter hybrid	Hybrid	Median = 0 Minimum = 0 Maximum = 41	Median = 0 Minimum = 0 Maximum = 26	Median = 1 Minimum = 0 Maximum = 67	Total = 9 Median = 0 Minimum = 0 Maximum = 2
Dragonflye hybrid	Hybrid	Median = 2.5 Minimum = 0 Maximum = 112	Median = 0 Minimum = 0 Maximum = 64	Median = 4.5 Minimum = 0 Maximum = 154	Total = 70 Median = 2 Minimum = 0 Maximum = 12
Unicycler	Hybrid	Median = 11 Minimum = 0 Maximum = 125	Median = 34 Minimum = 0 Maximum = 165	Median = 57.5 Minimum = 3 Maximum = 290	Total = 87 Median = 1 Minimum = 0 Maximum = 16
Hybracter long	Long	Median = 16 Minimum = 1 Maximum = 743	Median = 21.5 Minimum = 0 Maximum = 156	Median = 54 Minimum = 1 Maximum = 852	Total = 11 Median = 1 Minimum = 0 Maximum = 3
Dragonflye long	Long	Median = 125 Minimum = 2 Maximum = 4814	Median = 34.5 Minimum = 0 Maximum = 2172	Median = 170.5 Minimum = 2 Maximum = 6332	Total = 68 Median = 2 Minimum = 0 Maximum = 12

409

410

411 Similar results were found in the long-read only tool comparison. Hybracter long produced
412 the fewest SNVs (median 21.5) compared to Dragonflye long (median 34.5). Hybracter long
413 consistently had far fewer small InDels (median 16) and large InDels (total 11 across 30
414 samples) compared to Dragonflye long (median 125 and total 68 respectively). No perfect
415 chromosomes were assembled by either long-only tool, though Hybracter long did assemble
416 three near-perfect chromosomes (*L. monocytogenes* ATCC BAA-679 with simplex and
417 duplex super-accuracy model basecalled reads and ATCC 10708 *S. enterica* with duplex
418 super-accuracy model basecalled reads) and several chromosomes with fewer than 50 total
419 small InDels plus SNVs and 0 large InDels (*Lerminiaux* isolates A, G, H, L, J, and *S. aureus*
420 JKD6159 with R10 chemistry, ATCC 10708 *S. enterica* with simplex super-accuracy model
421 basecalled reads).
422
423 Overall, Hybracter long showed consistently worse performance than the hybrid tools
424 Hybracter hybrid and Dragonflye hybrid tools (though not Unicycler) as measured by SNVs
425 and small InDels. Combined with the lack of perfect assemblies even for duplex super-
426 accuracy model basecalled read assemblies, this suggests the continuing utility of short-read
427 polishing for the isolates surveyed.
428

429 Plasmid Recovery Performance and Accuracy

430
431 Hybracter in both hybrid and long modes was superior at recovering plasmids compared to
432 the other tools in the same class (Table 4). Hybracter hybrid was able to completely recover
433 65/69 possible plasmids (the other two were partially recovered), compared to 60/69 for
434 Unicycler and only 44/69 for Dragonflye hybrid. Hybracter hybrid did not miss a single

435 plasmid, while Unicycler missed 3/69 (all in Isolate E *Klebsiella pneumoniae* from
436 *Lerminiaux* et al.) and Dragonflye hybrid completely missed 9/69. In terms of plasmid
437 accuracy, Hybracter hybrid and Unicycler were similar in terms of SNVs plus small InDels,
438 with medians of 1.62 and 2.02 per 100kbp respectively (Supplementary Table 9), while
439 Hybracter hybrid produced fewer large InDels than Unicycler (44 vs 63 in total).
440
441 Interestingly, Hybracter long showed strong performance at recovering plasmids despite
442 using only long-reads, completely recovering 60/69 plasmids and completely missing only
443 4/69. This performance was far superior to Dragonflye long (44/69 completely recovered,
444 9/69 missed). In terms of accuracy, both long tools were similar and unsurprisingly less
445 accurate than the hybrid tools in terms of SNVs plus small InDels (medians of 8.74 per
446 100kbp for Hybracter long and 7.66 per 100kbp for Dragonflye long).
447
448 All five tools detected an additional 5411bp plasmid in *Lerminiaux* Isolate G not found in the
449 reference sequence and Hybracter in both hybrid and long modes detected a further 2519bp
450 small plasmid from this genome.
451
452 Hybracter hybrid recovers more plasmids than either Unicycler or Dragonflye because it uses
453 a dedicated plasmid assembler, Plassembler. In addition, Hybracter long, using only long-
454 reads had an identical complete plasmid recovery rate to Unicycler, which uses both long-
455 and short-reads (60/69 for both). These results suggest that Hybracter long, by applying
456 algorithms designed for short-reads on long-reads, largely solves the existing difficulties of
457 recovering small plasmids from long-reads, at least on the benchmarking dataset of
458 predominantly R10 Nanopore reads[19,51]. Even on the lower quality fast basecalled ATCC
459 reads, Hybracter long performed well, with only one sample failing to produce a plasmid

460 assembly similar to higher quality datasets (ATCC 10708 *S. enterica* – See Supplementary
461 Tables 6 and 7).
462
463 Another notable result from Hybracter hybrid is that in 10/30 samples, it assembled
464 additional non-plasmid contigs, which occurred in only 2/30 isolates for Unicycler. This is a
465 limitation of Hybracter hybrid, as the extra sensitivity to recover plasmids comes with the
466 cost of more false positive non-plasmid contigs that may be low-depth artefacts of
467 sequencing. Hybracter has a 'depth_filter' parameter (defaulting to 0.25x of the chromosome
468 depth) that filters out all non-circular putative plasmid contigs below this value.
469
470 It should be noted, however, that these contigs are not always an assembly artefact and can
471 provide additional information regarding the quality control and similarity of short and long-
472 read sets. In Plassembler implemented within Hybracter hybrid, the existence of such contigs
473 is often indicative of mismatches between long- and short-read sets[28], suggesting that there
474 may be some heterogeneity between long- and short-reads in those samples.

475 Table 4. The Total Number of Plasmids Recovered by Each Tool. There were 59 total
476 reference plasmids in the 30 samples.

Tool	Complete Plasmids Recovered	Total Plasmids Partially Recovered or Misassembled	Total Plasmids Missed	Additional Plasmids Recovered not in Reference	Samples with Additional Non-Plasmid Contigs Recovered
Hybracter hybrid	65	4	0	2	10
Unicycler	60	6	3	1	2

Dragonflye hybrid	44	16	9	1	10
Hybracter long	60	5	4	2	3
Dragonflye long	44	16	9	1	10

480 Runtime Performance Comparison

481

482 As shown in Table 5 and Figure 3, median wall-clock times with 8 threads for Dragonflye
483 hybrid (4m34s) were smaller than Hybracter hybrid (15m03s), which were in turn smaller
484 than Unicycler (50m25s). For the long-only tools, Dragonflye long (4m10s) was faster than
485 Hybracter long (11m46s). Hybracter long was consistently slightly faster than Hybracter
486 hybrid (Table 5).

487

488 The difference in runtime performance between Hybracter and Dragonflye is predominantly
489 the result of the included targeted plasmid assembly and the reorientation and assessment
490 steps in Hybracter that are not included in Dragonflye. Additionally, the results suggest
491 limited benefits to running Hybracter with more than eight threads. As explained in the
492 following section, if a user has multiple isolates to assemble, a superior approach is to modify
493 the configuration file specifying more efficient resource requirements for each job in
494 Hybracter.

495

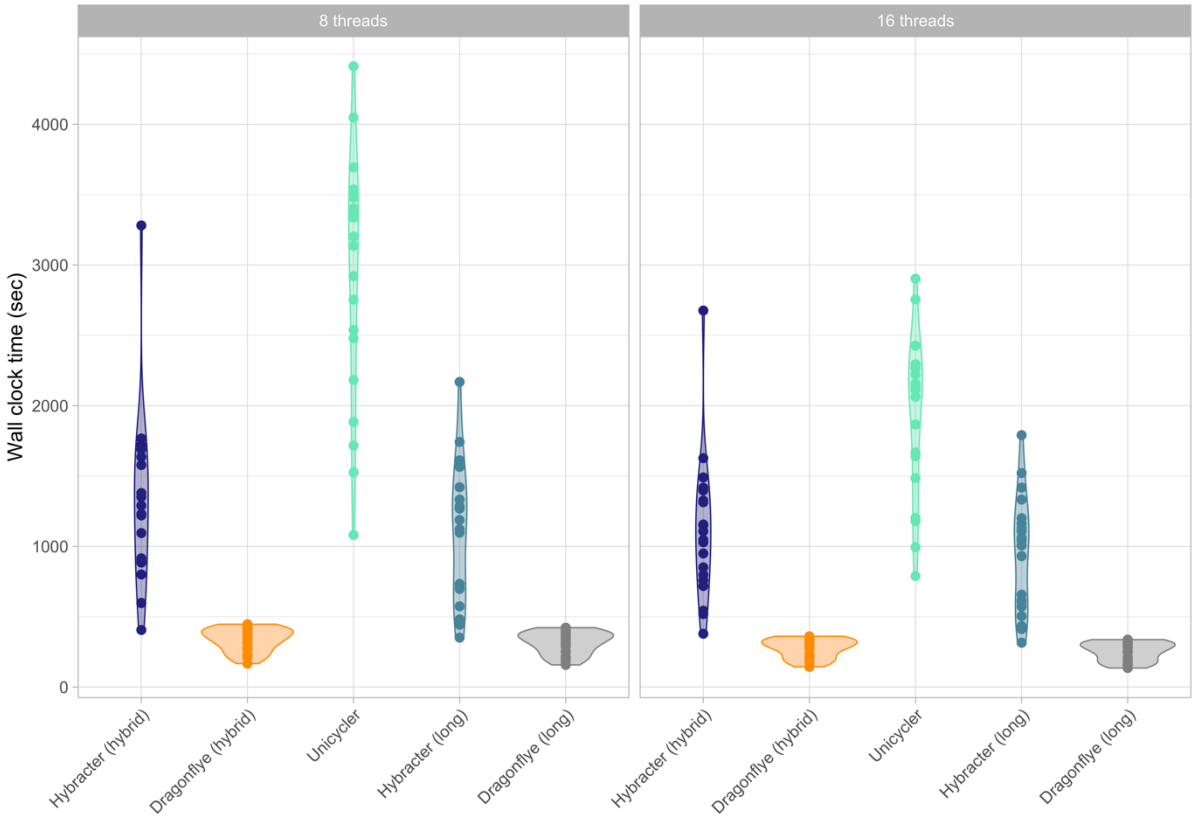
496 Table 5. Wall-clock Runtime Summary Statistics for Each Tool.

Tool	Type	8 Threads (hh:mm:ss)	16 Threads (hh:mm:ss)
Hybracter hybrid	Hybrid	Median = 00:15:03 Minimum = 00:04:29 Maximum = 00:54:41	Median = 00:13:44 Minimum = 00:03:27 Maximum = 00:44:36
Dragonflye hybrid	Hybrid	Median = 00:04:34 Minimum = 00:01:32 Maximum = 00:07:27	Median = 00:03:46 Minimum = 00:01:22 Maximum = 00:06:01
Unicycler	Hybrid	Median = 00:50:25 Minimum = 00:12:04 Maximum = 01:13:32	Median = 00:34:10 Minimum = 00:08:36 Maximum = 00:48:23
Hybracter long	Long	Median = 00:11:46 Minimum = 00:03:26 Maximum = 00:36:09	Median = 00:10:20 Minimum = 00:03:17 Maximum = 00:29:50
Dragonflye long	Long	Median = 00:04:10 Minimum = 00:01:22 Maximum = 00:06:01	Median = 00:04:34 Minimum = 00:01:32 Maximum = 00:07:27

497

498 Figure 3: Comparison of wall-clock runtime (in seconds) of Hybracter hybrid, Dragonflye

499 hybrid, Unicycler, Hybracter long and Dragonflye long when run with 8 and 16 threads.



500

501 Parallelisation Allows for Improved Efficiency

502

503 Hybracter allows users to specify and customise a configuration file to maximise resource

504 usage and runtime efficiency. Users can modify the desired threads, memory and time

505 requirements for each type of job that is run within Hybracter to suit their computational

506 resources. So that resources are not idle for most users on single sample assemblies, large

507 jobs such as the Flye and Plassembler assembly steps default to 16 threads and 32 GB of

508 memory.

509

510 To emphasise the efficiency benefits of parallelisation, the 12 *Lerminiaux* et al. isolates were
511 also assembled using 'hybracter hybrid' with a customised configuration file designed to
512 improve efficiency on the machine used for benchmarking. Specifically, the configuration
513 was changed to specify 8 threads and 16 GB of memory allocated to big jobs (assembly,
514 polishing and assessment) and 4 threads and 8 GB of memory allocated to medium jobs
515 (reorientation). More details on changing Hybracter's configuration file to suit specific
516 systems can be found in the documentation
517 (https://hybracter.readthedocs.io/en/latest/configuration/). We limited the overall 'hybracter
518 hybrid' run with 32 GB of memory and 16 threads to provide a fair comparison. The overall
519 'hybracter hybrid' run was then compared to the sum of the 12 'hybracter hybrid-single' runs.
520 Overall, the 12 isolates took 01h48m57s in the combined run, as opposed to 04h38m45s from
521 the sum of the 12 'hybracter hybrid-single' and 07h04m04s from the sum of the 12 Unicycler
522 runs. This inbuilt parallelisation of Hybracter provides significant efficiency benefits if
523 multiple samples are assembled simultaneously. The performance benefit of Hybracter
524 afforded by Snakemake integration in parallel computing systems may be variable over
525 different architectures, but this provides an example case of potential efficiency and
526 convenience benefits.
527

528 Long-Read Depth Does Not Affect Hybrid Assembly Accuracy If a Complete Chromosome
529 is Assembled

530 Finally, we tested the effect of long-read depth on the accuracy of assemblies with all five
531 tools at an estimated long-read depth from 10x to 100x at every interval of 5x for an example
532 isolate (*Lerminiaux* Isolate B, *Enterobacter cloacae*) with super-accuracy model basecalled
533 simplex reads (Figure 4 and Supplementary Table 12). At 10x and 15x sequencing depth,

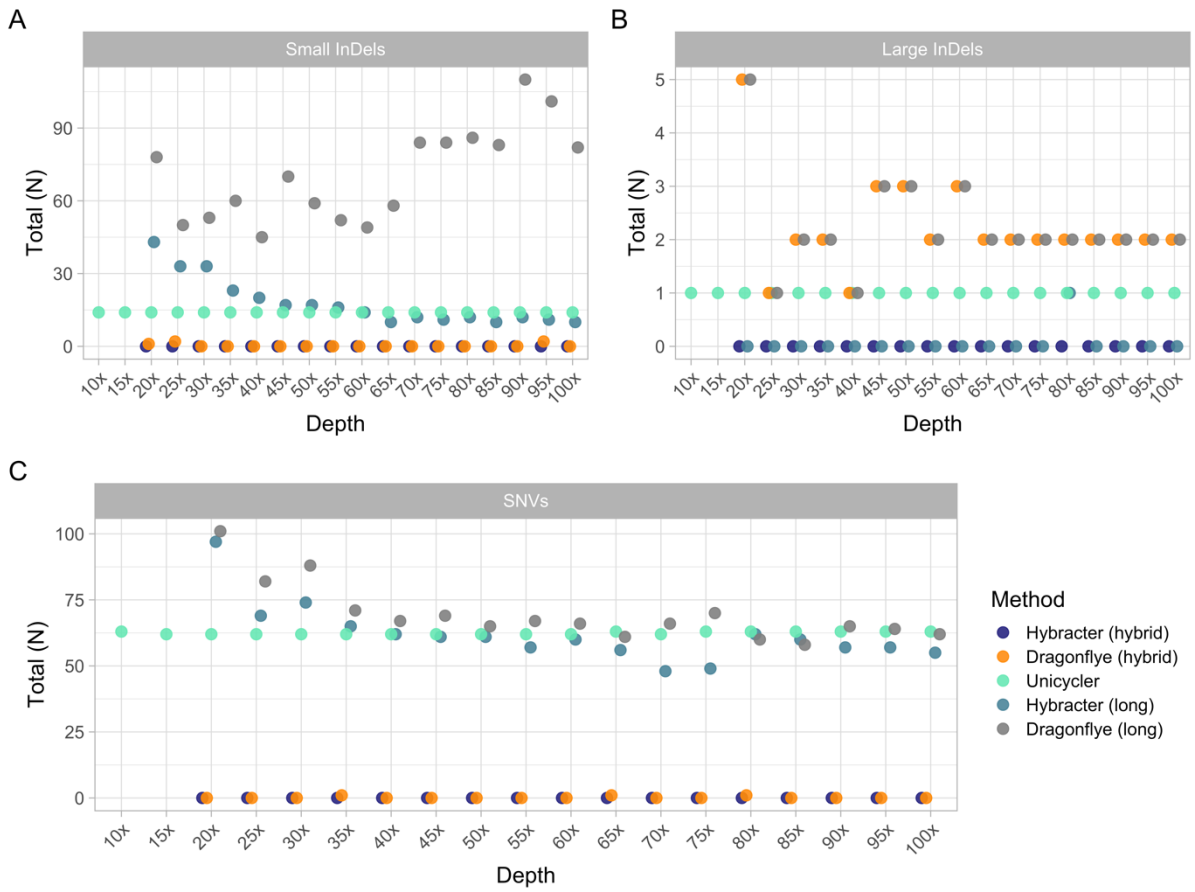
534 only Unicycler was able to assemble a complete chromosome. From 20x and above, all five
535 tools were able to assemble complete chromosomes. For the hybrid tools, once a complete
536 chromosome was assembled, increasing long-read depth had a negligible impact on accuracy
537 results (Figure 4). Notably, Hybracter hybrid was able to produce perfect assemblies from as
538 low as 20x long-read depth. For long-read only tools, increasing long-read depth did affect
539 accuracy. Increasing depth improved SNV accuracy for both Hybracter long and Dragonflye
540 long (Figure 4C). For small InDels, Hybracter long improved with extra depth, while
541 Dragonflye long actually performed worse (Figure 4A). Depth had minimal impact on large
542 InDels (Figure 4B).
543

544 Figure 4: Comparison of the counts of small (<60bp) (A) and large (>60bp) (B) insertions

545 and deletions (InDels) and SNVs (C) for Hybracter hybrid, Dragonflye hybrid, Unicycler,

546 Hybracter long and Dragonflye long chromosome assemblies of *Lerminiaux* Isolate B

547 (*Enterobacter cloacae*) at 5x intervals of sequencing depth from 10x to 100x.



548

549

550

551 ## Discussion

552

553 As long-read sequencing has improved in accuracy with reduced costs, it is now routine to
554 use a combination of long- and short-reads to generate complete bacterial genomes[3,5]. Recent
555 advances in assembly algorithms and accuracy improvements mean that a long-read first
556 hybrid assembly should be favoured with short-reads being used after assembly for polishing
557 [12], as opposed to the short-read first assembly approach (where long-reads are only used for
558 scaffolding a short-read assembly) utilised by the current gold standard automated assembler
559 Unicycler. The Unicycler approach is more prone to larger scale InDel errors as well as
560 smaller scale errors such as those caused by homopolymers or methylation motifs [6,11,52,53].
561 Additionally, it should be noted that it is already possible (while perhaps not routine) to
562 generate perfect hybrid bacterial genome assemblies using manual consensus approaches
563 requiring human intervention, such as Trycycler[7,54]. While manual approaches such as
564 Trycycler generally yield superior results to automated approaches, manually assembling
565 many complete genomes is challenging as considerable time, resources and bioinformatics
566 expertise are required.

567

568 The results of this study emphasise that the long-read first hybrid approach consistently
569 yields superior assemblies than the short-read first hybrid approach and should therefore be
570 preferred going forward. The only exception where a short-read first approach is to be
571 preferred is where a limited depth of long-read sequencing data is available (<20x depth). In
572 this instance, long-read first hybrid approaches may struggle to assemble a complete
573 chromosome, while short-read first approaches like Unicycler may be able to (Figure 4).

574

575 Interestingly, in the course of conducting benchmarking for this study, we found a large
576 number of discrepancies between older short-read first assembled 'reference genomes' for *S.*
577 *aureus* JKD6159 [55] and the five ATCC genomes benchmarked compared to updated
578 Trycycler long-read first references (See Supplementary Table 13). The number of
579 discrepancies ranged from 44 to 8,255 across the six genomes. Therefore, we recommend that
580 older short-read first reference genomes be updated if possible using a long-read assembly
581 approach (such as with Trycycler).
582
583 This study also shows that automated perfect hybrid genome assemblies are already possible
584 with Hybracter. This study and others[9,54] also confirm that a long-read first hybrid approach
585 remains preferable to long-read only assembly with Nanopore reads, as short-reads continue
586 to provide accuracy improvements in polishing steps. However, it is foreseeable that short-
587 reads will soon provide little or no accuracy improvements and will not be needed to polish
588 long-read only assemblies to perfection. Already, perfect long-read only assemblies are
589 possible, at least with manual intervention using Trycycler [7]. Accordingly, automated perfect
590 bacterial genome assemblies may soon become possible from long-reads only. Hybracter also
591 allows users to turn long-read polishing off altogether. It is already established that long-read
592 polishing can introduce errors and make long-read only assemblies worse with highly
593 accurate Nanopore and PacBio reads[11,31]. Therefore, this feature may become increasingly
594 useful as long-read sequencing continues to improve in accuracy and we recommend its use
595 for highly accurate Q20+ long-reads.
596
597 Hybracter was created to bridge the gap from the present to the future of automated perfect
598 hybrid and long-read-only bacterial genome assemblies. The results of this study show that
599 Hybracter in hybrid mode is both faster and more accurate than the current gold standard tool

600 for hybrid assembly Unicycler and is more accurate than Dragonflye in both modes. It should
601 be noted that if users want fast chromosome-only assemblies where accuracy is not essential
602 (for applications such as species identification or sequence typing), Dragonflye remains a
603 good option due to its speed.
604
605 Hybracter especially excels in recovering complete plasmid genomes compared to other
606 tools. By incorporating Plassembler, Hybracter recovers more complete plasmid genomes
607 than Unicycler in hybrid mode. Further, Hybracter long is comparable to Unicycler and
608 Hybracter hybrid when using long-reads only for plasmid recovery.
609
610 The high error rates of long-read sequencing technologies have prevented the application of
611 assembly approaches designed for highly accurate short-reads, such as constructing de Bruijn
612 graphs (DBGs) based on strings of a particular length *k* (*k*-mers) [56–58]. This resulted in
613 bioinformaticians initially utilising less efficient algorithms designed with long-reads in
614 mind, such as utilising overlap graphs in place of DBGs[27,37,39,59,60]. While DBGs have been
615 used for long-read assembly in some applications [61–63], adoption, especially in microbial
616 genomics, has been limited.
617
618 Although long-read first assembly methods enable complete chromosome and large plasmid
619 reconstruction, it is well established that long-read only assemblers struggle to assemble
620 small (<20kbp) plasmids accurately, often leading to missing or multiplicated assemblies[6,51].
621 These errors may be exacerbated if ligation chemistry-based sequencing kits are used[51].
622 Therefore, hybrid DBG based short-read first assemblies are traditionally recommended for
623 plasmid recovery[12].
624

625 Implemented in our post-publication changes to Plassembler described in this study,
626 Hybracter solves the problem of small plasmid recovery using long-reads. It achieves this by
627 implementing a DBG-based assembly approach with Unicycler. The same read set is used
628 twice, first as unpaired pseudo 'short' reads and then as long-reads; the long-read set
629 scaffolds a DBG-based assembly based on the same read set. This study demonstrates that
630 current long-read technologies, such as R10 Nanopore reads, are now accurate enough that
631 some short-read algorithms are applicable. Our results also suggest that similar DBG-based
632 algorithmic approaches could be used to enhance the recovery of small replicons in long-read
633 datasets beyond the use case presented here of plasmids in bacterial isolate assemblies. This
634 could potentially enhance the recovery of replicons such as bacteriophages[64] or other small
635 contigs from metagenomes using only long-reads[10,50].
636
637 Finally, consistent and resource efficient assemblies that are as accurate as possible in
638 recovering both plasmids and chromosomes are crucial, particularly for larger studies
639 investigating plasmid epidemiology and evolution. AMR genes carried on plasmids can have
640 complicated patterns of transmission involving horizontal transfer between different bacterial
641 species and lineages, transfer between different plasmid backbones, and integration into and
642 excision from the bacterial chromosome[65–67]. Accurate plasmid assemblies are crucial in
643 genomic epidemiology studies investigating transmission of antimicrobial resistant bacteria
644 within outbreak settings, as well as in a broader One Health context, where hundreds or even
645 thousands of assemblies may be analysed[68–71]. Hybracter will facilitate the expansion of such
646 studies, allowing for faster and more accurate automated complete genome assemblies than
647 existing tools. Additionally, by utilising Snakemake[20] with a Snaketool[21] command line
648 interface, Hybracter is easily and efficiently parallelised to optimise available resources over
649 various large-scale computing architectures. Individual jobs (such as each assembly,

reorientation, polishing or assessment step) within Hybracter are automatically sent to different resources on a high performance computing (HPC) cluster using the HPC's job scheduling system like Slurm[72]. Hybracter can natively use any Snakemake-supported cloud-based deployments such as Kubernetes, Google Cloud Life Sciences, Tibanna, and Azure Batch.

Conclusion

Hybracter is substantially faster than the current gold standard automated tool Unicycler, assembles chromosomes more accurately than existing methods, and is superior at recovering complete plasmid genomes. By applying DBG-based algorithms designed for short-reads on current generation long-reads, Hybracter long also solves the problem of long-read-only assemblers entirely missing or duplicating small circular elements such as plasmids. Hybracter is resource efficient and natively supports deployment on high-performance computer clusters and cloud environments for massively parallel analyses. We believe Hybracter will prove to be an extremely useful tool for the automated recovery of complete bacterial genomes from hybrid and long-read-only sequencing data suitable for massive datasets.

Acknowledgements

This work was supported with supercomputing resources provided by the Phoenix HPC service at the University of Adelaide. We would particularly like to thank Fabien Voisin for his integral role in maintaining and running Phoenix. We would also like to thank Brad Hart for useful comments in testing Hybracter and Simone Pignotti, Yu Wan and Oliver Schwengers for providing helpful comments and GitHub pull requests.

Funding

G.H. was supported by The University of Adelaide International Scholarships and a THRF Postgraduate Top-up Scholarship. A.E.S was supported by a University of Adelaide Barbara Kidman Women's Fellowship. R.A.E was supported by an award from the NIH NIDDK RC2DK116713 and an award from the Australian Research Council DP220102915. S.V. was supported by a Passe and Williams Foundation senior fellowship.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

690 ## References

691 1. Land, M. *et al.* Insights from 20 years of bacterial genome sequencing. *Funct Integr*
692 *Genomics* **15**, 141–161 (2015).
693 2. Goldstein, S., Beka, L., Graf, J. & Klassen, J. L. Evaluation of strategies for the assembly
694 of diverse bacterial genomes using MinION long-read sequencing. *BMC Genomics* **20**,
695 23 (2019).
696 3. De Maio, N. *et al.* Comparison of long-read sequencing technologies in the hybrid
697 assembly of complex bacterial genomes. *Microbial Genomics* **5**, e000294 (2019).
698 4. Wick, R. R., Judd, L. M., Gorrie, C. L. & Holt, K. E. Y. 2017. Completing bacterial
699 genome assemblies with multiplex MinION sequencing. *Microbial Genomics* **3**,
700 e000132.
701 5. Wick, R. R., Judd, L. M., Gorrie, C. L. & Holt, K. E. Unicycler: Resolving bacterial
702 genome assemblies from short and long sequencing reads. *PLOS Computational*
703 *Biology* **13**, e1005595 (2017).
704 6. Wick, R. R. & Holt, K. E. Benchmarking of long-read assemblers for prokaryote whole
705 genome sequencing. Preprint at https://doi.org/10.12688/f1000research.21782.4
706 (2021).
707 7. Wick, R. R. *et al.* Trycycler: consensus long-read assemblies for bacterial genomes.
708 *Genome Biology* **22**, 266 (2021).
709 8. Wick, R. ONT-only accuracy with R10.4.1. *Ryan Wick's bioinformatics blog*
710 https://rrwick.github.io/2023/05/05/ont-only-accuracy-with-r10.4.1.html
711 https://doi.org/10.5281/zenodo.7898220 (2023).
712 9. Lerminiaux, N., Fakharuddin, K., Mulvey, M. R. & Mataseje, L. Do we still need
713 Illumina sequencing data? Evaluating Oxford Nanopore Technologies R10.4.1 flow

714 cells and the Rapid v14 library prep kit for Gram negative bacteria whole genome
715 assemblies. *Can. J. Microbiol.* (2024) doi:10.1139/cjm-2023-0175.
716 10. Sereika, M. *et al.* Oxford Nanopore R10.4 long-read sequencing enables the
717 generation of near-finished bacterial genomes from pure cultures and metagenomes
718 without short-read or reference polishing. *Nat Methods* **19**, 823–826 (2022).
719 11. Wick, R. ONT-only accuracy: 5 kHz and Dorado. *Ryan Wick's bioinformatics blog*
720 https://rrwick.github.io/2023/10/24/ont-only-accuracy-update.html
721 https://doi.org/10.5281/zenodo.10038672 (2023).
722 12. Wick, R. R., Judd, L. M. & Holt, K. E. Assembling the perfect bacterial genome using
723 Oxford Nanopore and Illumina sequencing. *PLOS Computational Biology* **19**,
724 e1010905 (2023).
725 13. Murigneux, V. *et al.* MicroPIPE: validating an end-to-end workflow for high-quality
726 complete bacterial genome construction. *BMC Genomics* **22**, 474 (2021).
727 14. Schwengers, O. *et al.* ASA3P: An automatic and scalable pipeline for the assembly,
728 annotation and higher-level analysis of closely related bacterial isolates. *PLOS*
729 *Computational Biology* **16**, e1007134 (2020).
730 15. Petit, R. A. & Read, T. D. Bactopia: a Flexible Pipeline for Complete Analysis of
731 Bacterial Genomes. *mSystems* **5**, 10.1128/msystems.00190-20 (2020).
732 16. Petit III, R. A. Dragonflye: Assemble bacterial isolate genomes from Nanopore reads.
733 17. Hunt, M. *et al.* Circlator: automated circularization of genome assemblies using long
734 sequencing reads. *Genome Biology* **16**, 294 (2015).
735 18. Wick, R. R. & Holt, K. E. Polypolish: Short-read polishing of long-read bacterial
736 genome assemblies. *PLOS Computational Biology* **18**, e1009802 (2022).
737 19. Johnson, J., Soehnlen, M. & Blankenship, H. M. Long read genome assemblers
738 struggle with small plasmids. *Microbial Genomics* **9**, 001024 (2023).

739 20. Mölder, F. *et al.* Sustainable data analysis with Snakemake. Preprint at
740 https://doi.org/10.12688/f1000research.29032.2 (2021).
741 21. Roach, M. J. *et al.* Ten simple rules and a template for creating workflows-as-
742 applications. *PLOS Computational Biology* **18**, e1010705 (2022).
743 22. Wick, R. R. Filtlong. github. com/rrwick/Filtlong (2018).
744 23. Bonenfant, Q., Noé, L. & Touzet, H. Porechop_ABI: discovering unknown adapters in
745 Oxford Nanopore Technology sequencing reads for downstream trimming.
746 *Bioinformatics Advances* **3**, vbac085 (2023).
747 24. Roach, M. J. Trimnami: Trim lots of metagenomics samples all at once. (2023).
748 25. Chen, S., Zhou, Y., Chen, Y. & Gu, J. fastp: an ultra-fast all-in-one FASTQ preprocessor.
749 *Bioinformatics* **34**, i884–i890 (2018).
750 26. Shen, W., Le, S., Li, Y. & Hu, F. SeqKit: A Cross-Platform and Ultrafast Toolkit for
751 FASTA/Q File Manipulation. *PLOS ONE* **11**, e0163962 (2016).
752 27. Kolmogorov, M., Yuan, J., Lin, Y. & Pevzner, P. A. Assembly of long, error-prone reads
753 using repeat graphs. *Nat Biotechnol* **37**, 540–546 (2019).
754 28. Bouras, G., Sheppard, A. E., Mallawaarachchi, V. & Vreugde, S. Plassembler: an
755 automated bacterial plasmid assembly tool. *Bioinformatics* **39**, btad409 (2023).
756 29. medaka: Sequence correction provided by ONT Research.
757 30. Bouras, G., Grigson, S. R., Papudeshi, B., Mallawaarachchi, V. & Roach, M. J. Dnaapler:
758 A tool to reorient circular microbial genomes. *Journal of Open Source Software* **9**,
759 5968 (2024).
760 31. Bouras, G. *et al.* How low can you go? Short-read polishing of Oxford Nanopore
761 bacterial genome assemblies. 2024.03.07.584013 Preprint at
762 https://doi.org/10.1101/2024.03.07.584013 (2024).

763 32. Zimin, A. V. & Salzberg, S. L. The genome polishing tool POLCA makes fast and
764 accurate corrections in genome assemblies. *PLOS Computational Biology* **16**,
765 e1007981 (2020).
766 33. Clark, S. C., Egan, R., Frazier, P. I. & Wang, Z. ALE: a generic assembly likelihood
767 evaluation framework for assessing the accuracy of genome and metagenome
768 assemblies. *Bioinformatics* **29**, 435–443 (2013).
769 34. Larralde, M. Pyrodigal: Python bindings and interface to Prodigal, an efficient
770 method for gene prediction in prokaryotes. *Journal of Open Source Software* **7**, 4296
771 (2022).
772 35. Hyatt, D. *et al.* Prodigal: prokaryotic gene recognition and translation initiation site
773 identification. *BMC Bioinformatics* **11**, 119 (2010).
774 36. Vaser, R. & Šikić, M. Time- and memory-efficient genome assembly with Raven. *Nat*
775 *Comput Sci* **1**, 332–336 (2021).
776 37. Ruan, J. & Li, H. Fast and accurate long-read assembly with wtdbg2. *Nat Methods* **17**,
777 155–158 (2020).
778 38. Li, H. Minimap and miniasm: fast mapping and de novo assembly for noisy long
779 sequences. *Bioinformatics* **32**, 2103–2110 (2016).
780 39. Koren, S. *et al.* Canu: scalable and accurate long-read assembly via adaptive k-mer
781 weighting and repeat separation. *Genome Res.* **27**, 722–736 (2017).
782 40. Zhang, X. *et al.* Benchmarking of long-read sequencing, assemblers and polishers for
783 yeast genome. *Briefings in Bioinformatics* **23**, bbac146 (2022).
784 41. Vaser, R., Sović, I., Nagarajan, N. & Šikić, M. Fast and accurate de novo genome
785 assembly from long uncorrected reads. *Genome Res.* **27**, 737–746 (2017).
786 42. Chitale, P. *et al.* A comprehensive update to the Mycobacterium tuberculosis H37Rv
787 reference genome. *Nat Commun* **13**, 7068 (2022).

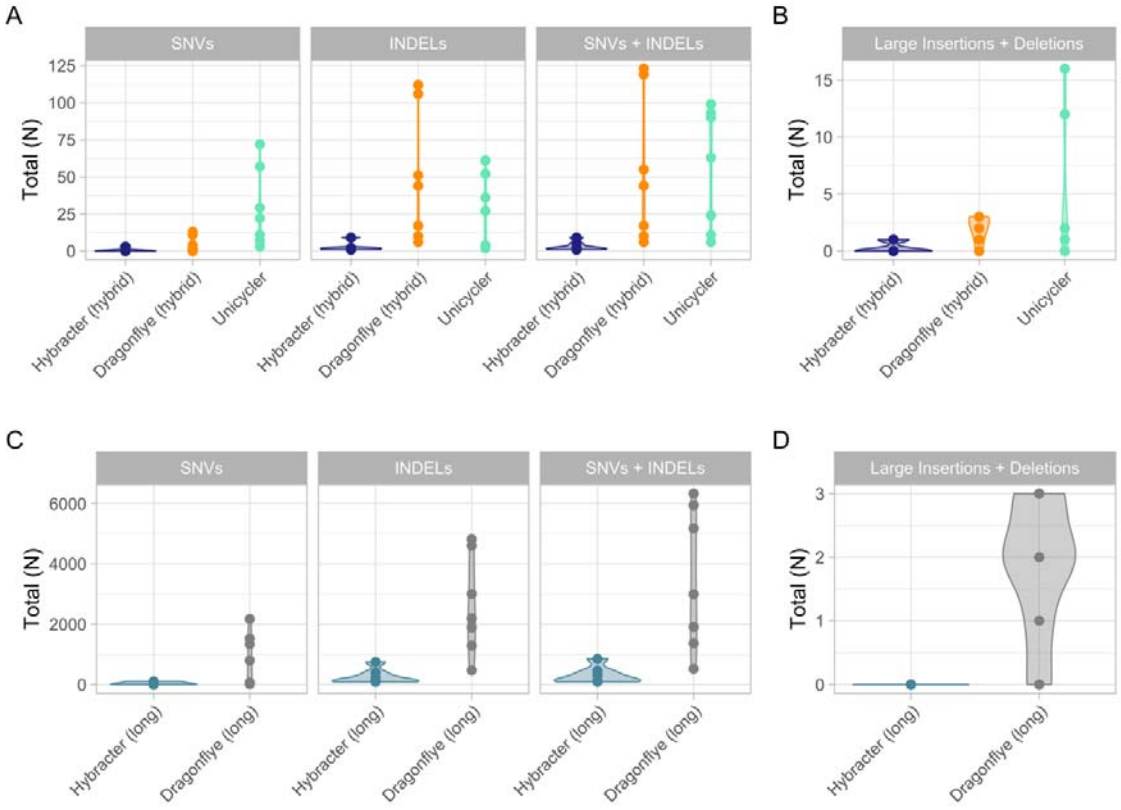
788 43. Hall, M. B. Rasusa: Randomly subsample sequencing reads to a specified coverage.
789 *Journal of Open Source Software* **7**, 3941 (2022).
790 44. Steinig, E. & Coin, L. Nanoq: ultra-fast quality control for nanopore reads. *Journal of*
791 *Open Source Software* **7**, 2991 (2022).
792 45. Hall, M. B. *et al.* Benchmarking reveals superiority of deep learning variant callers on
793 bacterial nanopore sequence data. 2024.03.15.585313 Preprint at
794 https://doi.org/10.1101/2024.03.15.585313 (2024).
795 46. Wick, R. R., Judd, L. M., Monk, I. R., Seemann, T. & Stinear, T. P. Improved Genome
796 Sequence of Australian Methicillin-Resistant Staphylococcus aureus Strain JKD6159.
797 *Microbiology Resource Announcements* **12**, e01129-22 (2023).
798 47. Kurtz, S. *et al.* Versatile and open software for comparing large genomes. *Genome*
799 *Biology* **5**, R12 (2004).
800 48. Sayers, E. W. *et al.* Database resources of the national center for biotechnology
801 information. *Nucleic Acids Research* **50**, D20–D26 (2022).
802 49. Galata, V., Fehlmann, T., Backes, C. & Keller, A. PLSDB: a resource of complete
803 bacterial plasmids. *Nucleic Acids Research* **47**, D195–D202 (2019).
804 50. Kolmogorov, M. *et al.* metaFlye: scalable long-read metagenome assembly using
805 repeat graphs. *Nat Methods* **17**, 1103–1110 (2020).
806 51. Wick, R. R., Judd, L. M., Wyres, K. L. & Holt, K. E. Y. 2021. Recovery of small plasmid
807 sequences via Oxford Nanopore sequencing. *Microbial Genomics* **7**, 000631 (2021).
808 52. Marinus, M. G. & Løbner-Olesen, A. DNA Methylation. *EcoSal Plus* **6**,
809 10.1128/ecosalplus.ESP-0003–2013 (2014).
810 53. Wick, R. R., Judd, L. M. & Holt, K. E. Performance of neural network basecalling tools
811 for Oxford Nanopore sequencing. *Genome Biology* **20**, 129 (2019).

812 54. Sanderson, N. D. *et al.* Comparison of R9.4.1/Kit10 and R10/Kit12 Oxford Nanopore
813 flowcells and chemistries in bacterial genome reconstruction. *Microbial Genomics* **9**,
814 000910 (2023).
815 55. Chua, K. *et al.* Complete Genome Sequence of Staphylococcus aureus Strain JKD6159,
816 a Unique Australian Clone of ST93-IV Community Methicillin-Resistant
817 Staphylococcus aureus. *Journal of Bacteriology* **192**, 5556–5557 (2010).
818 56. Bankevich, A. *et al.* SPAdes: A New Genome Assembly Algorithm and Its Applications
819 to Single-Cell Sequencing. *Journal of Computational Biology* **19**, 455–477 (2012).
820 57. Li, D., Liu, C.-M., Luo, R., Sadakane, K. & Lam, T.-W. MEGAHIT: an ultra-fast single-
821 node solution for large and complex metagenomics assembly via succinct de Bruijn
822 graph. *Bioinformatics* **31**, 1674–1676 (2015).
823 58. Compeau, P. E. C., Pevzner, P. A. & Tesler, G. How to apply de Bruijn graphs to
824 genome assembly. *Nature Biotechnology* **29**, 987–991 (2011).
825 59. Wong, J. *et al.* Linear time complexity de novo long read genome assembly with
826 GoldRush. *Nat Commun* **14**, 2906 (2023).
827 60. Amarasinghe, S. L. *et al.* Opportunities and challenges in long-read sequencing data
828 analysis. *Genome Biology* **21**, 30 (2020).
829 61. Ekim, B., Berger, B. & Chikhi, R. Minimizer-space de Bruijn graphs: Whole-genome
830 assembly of long reads in minutes on a personal computer. *Cell Syst* **12**, 958-968.e6
831 (2021).
832 62. Bankevich, A., Bzikadze, A. V., Kolmogorov, M., Antipov, D. & Pevzner, P. A. Multiplex
833 de Bruijn graphs enable genome assembly from long, high-fidelity reads. *Nat
834 Biotechnol* **40**, 1075–1081 (2022).
835 63. Lin, Y. *et al.* Assembly of long error-prone reads using de Bruijn graphs. *Proceedings
836 of the National Academy of Sciences* **113**, E8396–E8405 (2016).

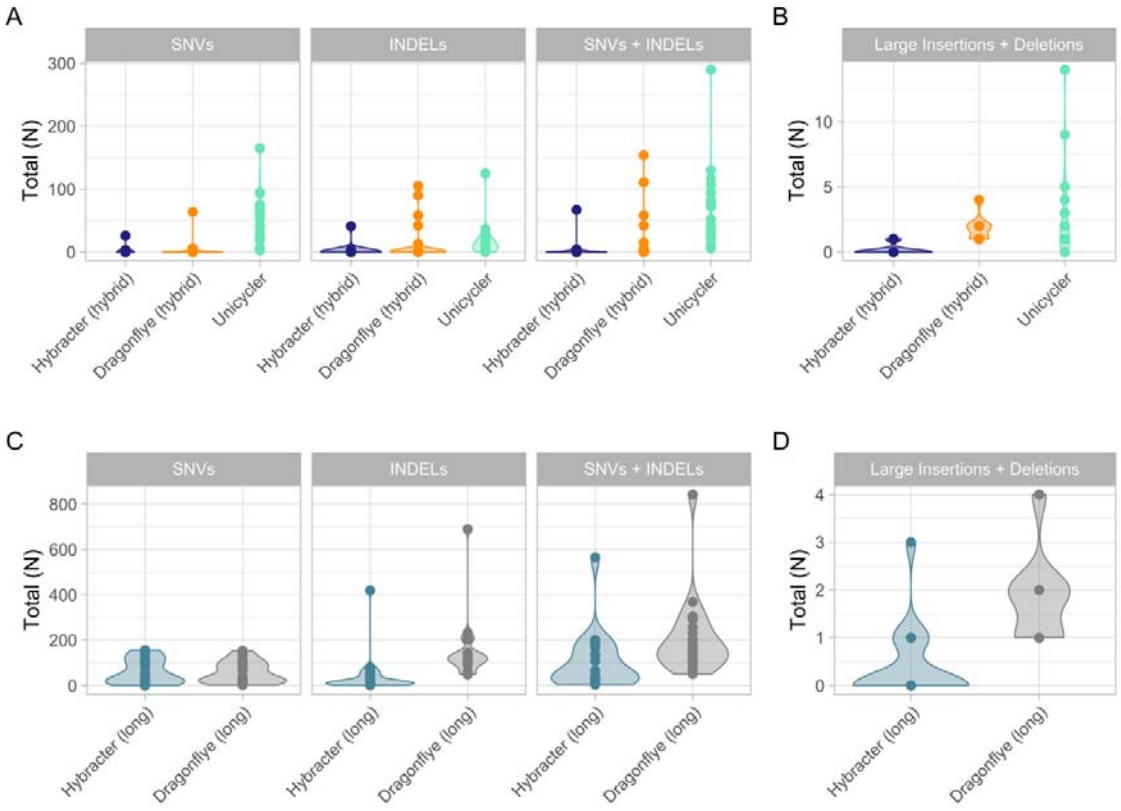
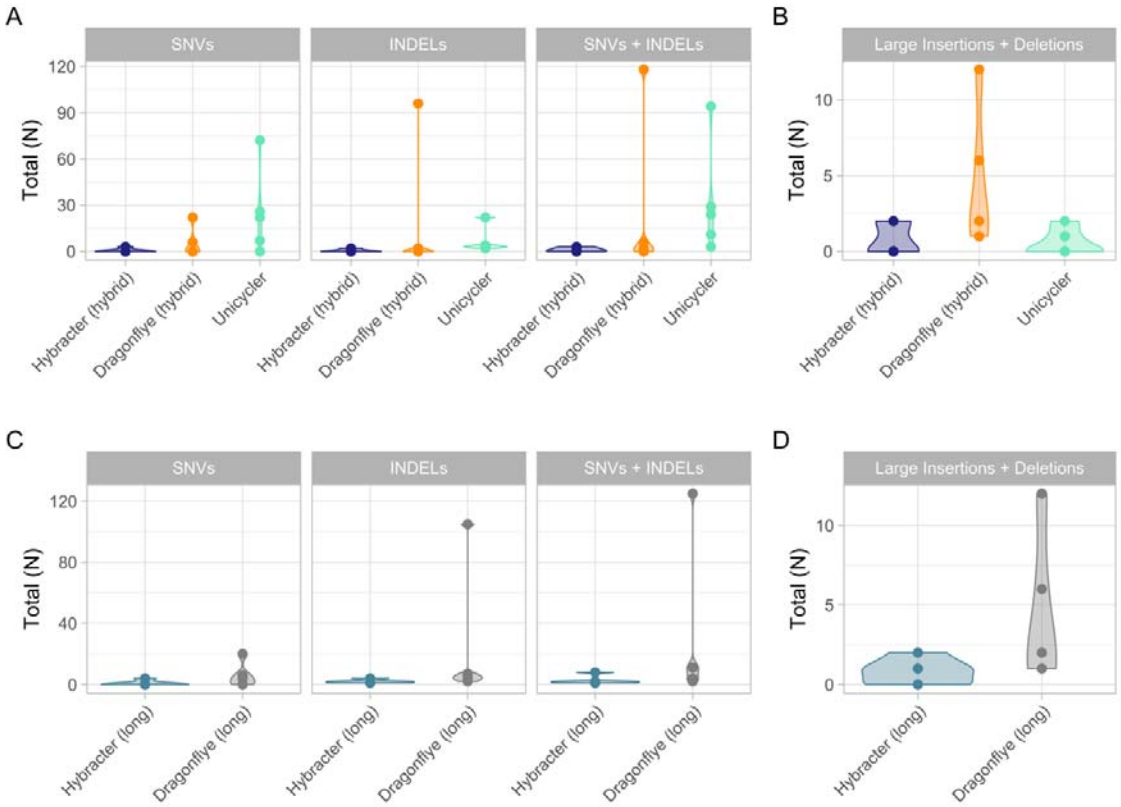
837 64. Mallawaarachchi, V. *et al.* Phables: from fragmented assemblies to high-quality
838 bacteriophage genomes. *Bioinformatics* **39**, btad586 (2023).
839 65. Mathers, A. J. *et al.* Chromosomal Integration of the Klebsiella pneumoniae
840 Carbapenemase Gene, blaKPC, in Klebsiella Species Is Elusive but Not Rare.
841 *Antimicrobial Agents and Chemotherapy* **61**, e01823-16 (2017).
842 66. Houtak, G. *et al.* The intra-host evolutionary landscape and pathoadaptation of
843 persistent Staphylococcus aureus in chronic rhinosinusitis. *Microbial Genomics* **9**,
844 001128 (2023).
845 67. Sheppard, A. E. *et al.* Nested Russian Doll-Like Genetic Mobility Drives Rapid
846 Dissemination of the Carbapenem Resistance Gene blaKPC. *Antimicrobial Agents and*
847 *Chemotherapy* **60**, 3767–3778 (2016).
848 68. Hawkey, J. *et al.* ESBL plasmids in Klebsiella pneumoniae: diversity, transmission
849 and contribution to infection burden in the hospital setting. *Genome Medicine* **14**, 97
850 (2022).
851 69. Matlock, W. *et al.* Enterobacterales plasmid sharing amongst human bloodstream
852 infections, livestock, wastewater, and waterway niches in Oxfordshire, UK. *eLife* **12**,
853 e85302 (2023).
854 70. Roberts, L. W. *et al.* Long-read sequencing reveals genomic diversity and associated
855 plasmid movement of carbapenemase-producing bacteria in a UK hospital over 6
856 years. *Microbial Genomics* **9**, 001048 (2023).
857 71. Lerminiaux, N. *et al.* Plasmid genomic epidemiology of blaKPC carbapenemase-
858 producing Enterobacterales in Canada, 2010–2021. *Antimicrobial Agents and*
859 *Chemotherapy* **0**, e00860-23 (2023).
860 72. Yoo, A. B., Jette, M. A. & Grondona, M. SLURM: Simple Linux Utility for Resource
861 Management. in *Job Scheduling Strategies for Parallel Processing* (eds. Feitelson, D.,

Rudolph, L. & Schwiegelshohn, U.) 44–60 (Springer, Berlin, Heidelberg, 2003). doi:10.1007/10968987_3.

Supplementary Figure 1: For lower quality samples only (all five ATCC fast model base-called samples, *S. aureus* JKD6159 with R9 chemistry and *M. tuberculosis* H37R2), this figure presents a comparison of the counts of small nucleotide variants (SNVs) and small (<60bp) insertions and deletions (InDels) (A) and the total number of large (>60bp) InDels (B) for the hybrid tools benchmarked (Hybracter hybrid in dark blue, Dragonflye hybrid in orange and Unicycler in green). The counts of SNVs and small InDels (C) and the total number of large InDels (D) for the long tools benchmarked (Hybracter long in light blue, Dragonflye long in grey) are also shown. All data presented is from the benchmarking output run with 8 threads.



Supplementary Figure 2: For simplex super-accuracy model basecalled samples (five ATCC, *S. aureus* JKD6159 with R10 chemistry and all twelve *Lerminiaux* et al. samples), this figure presents a comparison of the counts of small nucleotide variants (SNVs) and small (<60bp) insertions and deletions (InDels) (A) and the total number of large (>60bp) InDels (B) for the hybrid tools benchmarked (Hybracter hybrid in dark blue, Dragonflye hybrid in orange and Unicycler in green). The counts of SNVs and small InDels (C) and the total number of large InDels (D) for the long tools benchmarked (Hybracter long in light blue, Dragonflye long in grey) are also shown. All data presented is from the benchmarking output run with 8 threads.



Supplementary Figure 3: For duplex super-accuracy model basecalled samples (five ATCC samples), this figure presents a comparison of the counts of small nucleotide variants (SNVs) and small (<60bp) insertions and deletions (InDels) (A) and the total number of large (>60bp) InDels (B) for the hybrid tools benchmarked (Hybracter hybrid in dark blue, Dragonflye hybrid in orange and Unicycler in green). The counts of SNVs and small InDels (C) and the total number of large InDels (D) for the long tools benchmarked (Hybracter long in light blue, Dragonflye long in grey) are also shown. All data presented is from the benchmarking output run with 8 threads.