Bioinformatics, 2016, 1–8
doi: 10.1093/bioinformatics/btw237
Advance Access Publication Date: 9 May 2016
Original Paper

OXFORD

Genome analysis

Evaluation of hybrid and non-hybrid methods for *de novo* assembly of nanopore reads

Ivan Sović[1,†], Krešimir Križanović[2,†], Karolj Skala[1] and Mile Šikić[2,3,*]

[1]Centre for Informatics and Computing, Ruđer Bošković Institute, 10000 Zagreb, Croatia, [2]Department of Electronic Systems and Information Processing, Faculty of Electrical Engineering and Computing, University of Zagreb, 10000 Zagreb, Croatia and [3]Bioinformatics Institute, Singapore 138671, Singapore

*To whom correspondence should be addressed.
†These authors contributed equally to this study.
Associate Editor: Inanc Birol

Received on November 20, 2015; revised on March 14, 2016; accepted on April 25, 2016

Abstract

Motivation: Recent emergence of nanopore sequencing technology set a challenge for established assembly methods. In this work, we assessed how existing hybrid and non-hybrid *de novo* assembly methods perform on long and error prone nanopore reads.
Results: We benchmarked five non-hybrid (in terms of both error correction and scaffolding) assembly pipelines as well as two hybrid assemblers which use third generation sequencing data to scaffold Illumina assemblies. Tests were performed on several publicly available MinION and Illumina datasets of *Escherichia coli* K-12, using several sequencing coverages of nanopore data (20×, 30×, 40× and 50×). We attempted to assess the assembly quality at each of these coverages, in order to estimate the requirements for closed bacterial genome assembly. For the purpose of the benchmark, an extensible genome assembly benchmarking framework was developed. Results show that hybrid methods are highly dependent on the quality of NGS data, but much less on the quality and coverage of nanopore data and perform relatively well on lower nanopore coverages. All non-hybrid methods correctly assemble the *E. coli* genome when coverage is above 40×, even the non-hybrid method tailored for Pacific Biosciences reads. While it requires higher coverage compared to a method designed particularly for nanopore reads, its running time is significantly lower.
Availability and Implementation: https://github.com/kkrizanovic/NanoMark
Contact: mile.sikic@fer.hr
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

During the last ten years next generation sequencing (NGS) devices have dominated genome sequencing market. In contrast to previously used Sanger sequencing, NGS is much cheaper, less time consuming and not so labor intensive. Yet, when it comes to *de novo* assembly of longer genomes many researchers are being skeptical of using NGS reads. These devices produce reads a few hundred base pairs long, which is too short to unambiguously resolve repetitive regions even within relatively small microbial genomes (Nagarajan and Pop, 2013).

Although the use of paired-end and mate-pair technologies has improved the accuracy and completeness of assembled genomes, NGS sequencing still produces highly fragmented assemblies due to long repetitive regions. These incomplete genomes have to be finished using a more laborious approach that includes Sanger sequencing and specially tailored assembly methods. Owing to NGS, many efficient algorithms have been developed to optimize running time and memory footprints in sequence assembly, alignment and downstream analysis steps.

The need for technologies that would produce longer reads which could solve the problem of repeating regions has resulted in the advent of new sequencing approaches—the so-called 'third generation sequencing technologies'. The first among them was a single-molecule sequencing technology developed by Pacific Biosciences (PacBio). Although PacBio sequencers produce much longer reads (up to several tens of thousands of base pairs), their reads have significantly higher error rate (∼10 to 15%) than NGS reads (∼1%) (Schirmer *et al.*, 2015). Existing assembly and alignment algorithms were not capable of handling such high error rates. This caused the development of read error correction methods. At first, hybrid correction was performed using complementary NGS (Illumina) data (Koren *et al.*, 2012). Later, self-correction of PacBio-only reads was developed (Chin *et al.*, 2013) which required higher coverage (>50×). The development of new, more sensitive aligners (BLASR (Chaisson and Tesler, 2012)) and optimization of existing ones (BWA-MEM (Li, 2013)) was required.

In 2014, Oxford Nanopore Technologies (ONT) presented their tiny MinION sequencer—about the size of a harmonica. The MinION can produce reads up to a few hundred thousand base pairs long.1D reads from the MinION sequencer (with the latest R7.3 chemistry) have raw base accuracy less than 75%, while higher quality 2D reads (80–88% accuracy) comprise only a fraction of all 2D reads (Ip *et al.*, 2015; Laver *et al.*, 2015). This again spurred the need for development of even more sensitive algorithms for mapping and realignment, such as GraphMap (Sović *et al.*, 2016) and marginAlign (Jain *et al.*, 2015). Any doubt about the possibility of using MinION reads for *de novo* assembly was resolved in 2015 when Loman *et al.* demonstrated that the assembly of a bacterial genome (*Escherichia coli* K-12) using solely ONT reads is possible even with high error rates (Loman *et al.*, 2015). Thanks to extremely long reads and affordability and availability of the nanopore sequencing technology, these results might cause a revolution in *de novo* sequence analysis in near future.

Following up on the results from Loman *et al.* and Liao *et al.* (2015), we explored the applicability of existing hybrid and non-hybrid *de novo* assembly tools that support third generation sequencing data and assessed their ability to cope with nanopore error profiles. In our study, we compared seven assembly tools/pipelines which include five long-read assemblers: pipeline published by Loman *et al.* (in continuation LQS pipeline), PBcR (Koren *et al.*, 2012), Falcon (https://github.com/PacificBiosciences/FALCON), Miniasm (Li, 2016) and Canu (https://github.com/marbl/canu); and two hybrid assemblers: ALLPATHS-LG (Gnerre *et al.*, 2011) and SPAdes (Bankevich *et al.*, 2012). These tools were tested on real, publicly available datasets of a well-known clonal sample of *E. coli* K-12 MG1655. All of the tools/pipelines were evaluated on all test datasets up to the draft assembly level, not including the polishing phase. Draft assemblies, containing one 'big contig' of at least 4 Mbp, were polished using Nanopolish and compared. For the purpose of our analyses, we developed a benchmarking framework NanoMark which automates running of different assemblers and processing of the results. The framework provides wrappers with uniform interfaces for each assembly tool, simplifying their usage for the end user.

2 Background

Majority of algorithms for *de novo* assembly follow either the de Bruijn graph (DBG) or the Overlap-Layout-Consensus (OLC) paradigm (Pop, 2009). OLC assemblers predate the DBG and were widely used in the Sanger sequencing era. A major representative of the OLC class is Celera which was developed and maintained until very recently. The DBG approach attempted to solve the problem of ever-growing sequencing throughput brought on by the NGS technologies. Unlike OLC in which overlaps between reads have to be calculated explicitly, DBG splits the reads into k-mers and constructs the overlap graph implicitly, e.g. through a hash table lookup. While the assembly in the OLC paradigm attempts to find a Hamiltonian path through an overlap graph, the DBG attempts to solve a, virtually, simpler problem of finding an Eulerian path through a de Bruijn graph. It was later shown that both de Bruijn and overlap graphs can be transformed into string graph form, in which, similar to the DBG, an Eulerian path also needs to be found to obtain the assembly (Myers, 2005). Major differences lie in the implementation specifics of both algorithms. Although the DBG approach is faster, OLC based algorithms perform better for longer reads (Pop, 2009). Additionally, DBG assemblers depend on finding exact-matching k-mers between reads (typically ∼21 to 127 bases long (Bankevich and Pevzner, 2016)). Given the error rates in third generation sequencing data, this presents a serious limitation. The OLC approach, on the other hand, should be able to cope with higher error rates given a sensitive enough overlapper, but contrary to the DBG a time-consuming all-to-all pairwise comparison between input reads needs to be performed.

Since the focus in the past decade has been on NGS reads, most of the state-of-the-art assemblers use the DBG paradigm. Hence, there are not many OLC assemblers that could be utilized for long PacBio and ONT reads. In fact, methods developed to handle such data are mostly pipelines based on the Celera assembler, including: HGAP (Chin *et al.*, 2013), PBcR (Koren *et al.*, 2012) and the LQS pipeline (Loman *et al.*, 2015). Since its original publication (Myers *et al.*, 2000), Celera has been heavily revised to support newer sequencing technologies, including modifications for second generation data (Miller *et al.*, 2008), adoptions for third generation (single molecule) data via hybrid error correction (Koren *et al.*, 2012), non-hybrid error correction (Berlin *et al.*, 2015; Miller *et al.*, 2008) and hybrid approaches to assembly which combine two or more technologies (Goldberg *et al.*, 2006). All of this contributed to the popularity of Celera which led to its wide adoption in assembly pipelines for third generation sequencing data. Notably, one of the first such pipelines was the Hierarchical Genome Assembly Process (HGAP). HGAP uses BLASR to detect overlaps between raw reads during the error correction step. Unfortunately, it requires input data to be in PacBio-specific formats, which prevents its application to other (e.g. nanopore) sequencing technologies. PBcR pipeline employs a similar approach to HGAP—it starts with an error correction step, and feeds Celera with corrected reads. PBcR, since recently, employs the MHAP overlapper (Berlin *et al.*, 2015) for sensitive overlapping of reads during the error–correction step. Also, recent updates allow it to handle reads from Oxford Nanopore MinION sequencers. The LQS pipeline also follows a similar approach, but implements novel error–correction (Nanocorrect) and consensus (Nanopolish) steps. Instead of BLASR and MHAP, Nanocorrect uses DALIGNER (Myers, 2014) for overlap detection. Nanopolish presents a new signal-level consensus method for fine-polishing of a draft assembly using raw nanopore data. The LQS pipeline also employs Celera as the middle layer, i.e. for assembly of error corrected reads.

Until very recently, the only non-hybrid alternative to Celera-based pipelines was Falcon. Falcon is a new experimental diploid assembler developed by Pacific Biosciences, not yet officially published. It is based on a hierarchical approach similar to HGAP,

consisting of several steps: (i) raw sub-read overlapping for error correction using DALIGNER, (ii) pre-assembly and error correction, (iii) overlapping of error-corrected reads, (iv) filtering of overlaps, (v) construction of the string graph and (vi) contig construction. Unlike HGAP, it does not use Celera as its core assembler. Since Falcon accepts input reads in the standard FASTA format and not only the PacBio-specific format like HGAP does, it can potentially be used on any base called long-read dataset. Although originally intended for PacBio data, Falcon presents a viable option for assembly of nanopore reads, even though they have notably different error profiles.

In late 2015 the developers of Celera, PBcR and MHAP moved away from original Celera and PBcR projects and started to develop a new assembler, Canu. Canu is derived from Celera and also utilizes code from Pacific Biosciences' Falcon and Pbdagcon projects. It is still in the very early phase of development.

Also in late 2015, a new long read assembly tool Miniasm was released (Li, 2016). Miniasm attempts to assemble genomes from noisy long reads (both PacBio and Oxford Nanopore) without performing error–correction.

Aside from mentioned methods, hybrid assembly approaches present another avenue to utilizing nanopore sequencing data. Liao et al. (Liao *et al.*, 2015) recently evaluated several assembly tools on PacBio data, including hybrid assemblers SPAdes (Bankevich *et al.*, 2012) and ALLPATHS-LG (Gnerre *et al.*, 2011) for which they reported good results. Both of these are DBG-based, use Illumina libraries for the primary assembly and then attempt to scaffold the assemblies using longer, less accurate reads. Furthermore, SPAdes was recently updated and now officially supports nanopore sequencing data as the long read complement to NGS data.

3 Methods

Since, to the best of our knowledge, no dedicated MinION read simulator exists, we focused our benchmark on real nanopore sequencing datasets. Although there is a number of publicly available datasets, many of them consist either of organisms/strains which do not yet have officially finished genome assemblies, or the coverage of the dataset is not high enough to provide informative nanopore-only assembly results. Aside from the Lambda phage (which comes as a burn-in sample for every MinION), sequencing data for the well-known clonal sample of *E. coli* K-12 MG1655 are the most abundant. In this study, we use several most recent *E. coli* K-12 datasets to reflect on the current state of the nanopore data as well as the quality of assembly they provide. In addition to using entire datasets, we subsampled two of the datasets to provide a larger span of coverages in order to inspect the scalability of assemblers as well as their ability to cope with the abundance or the lack of data.

3.1 Datasets

Benchmarking datasets were extracted from several publicly available nanopore datasets and one publicly available Illumina dataset. These are:

- **ERX708228, ERX708229, ERX708230, ERX708231:** four flowcells used in Loman *et al.* nanopore assembly paper (Loman *et al.*, 2015).
- ***E. coli* K-12 MG1655 R7.3 dataset** (Quick *et al.*, 2014).
- **MARC, WTCHG dataset** (Ip *et al.*, 2015)**:** A dataset recently published by the MinION Analysis and Reference Consortium, consists of a compilation of data generated using several MinION sequencers in laboratories distributed world-wide.
- ***E. coli* K-12 MG1655 SQK-MAP006-1 dataset:** This is the most recent publicly available MinION dataset, obtained using the newest sequencing protocol. Link: http://lab.loman.net/2015/09/24/first-sqk-map-006-experiment/
- **Illumina frag and jump libraries** (Liao *et al.*, 2015): Link: ftp://ftp.broadinstitute.org/pub/papers/assembly/Ribeiro2012/data/ecoli_data_alt.tar.gz

The test datasets were designed with the idea to test the effect of varying coverage and data quality on the assembly process, and consist of either full datasets described above or subsampled versions of these datasets. Test datasets used for benchmarking are presented in Table 1.

For each of the benchmarking datasets we analyzed the error rates present in the data (Supplementary Table 1). All reads were aligned to the *E. coli* K-12 reference (NC_000913.3) using GraphMap (parameters '-a anchorgotoh'). Analysis shows a clear distinction between older (Dataset 1: 33% error rate) and newer (Dataset 2 and 3: 16%, Dataset 4: 11% and Dataset 5: 10% error rates) nanopore data, as well as Illumina data (3% error rate).

3.2 Data preparation

For nanopore datasets, sequence data was extracted from basecalled FAST5 files using Poretools (Loman and Quinlan, 2014). For Datasets 1, 3, 4 and 5 the entire set of reads was extracted and used for analyses. For Dataset 2, only flowcells ERX708228, ERX708229 and ERX708230 were used to obtain coverage close to 20×. Datasets 1 was prepared for testing the assemblers' robustness on 1d reads. Additionally, Dataset 5 (∼50× coverage) was subsampled to four different coverages: 32×, 35×, 37× and 40×. This was done in order to enable a more fine-grained estimation of the amount of sequencing data required for a complete genome assembly.

Hybrid assemblers were tested using the Illumina dataset together with each nanopore test dataset. They were also run on the Illumina dataset alone, to get a reference for the assembly quality and to be able to estimate the contribution to the assembly when nanopore reads are added. All libraries in the Illumina dataset come with reads and quality values in separate files (fasta and quala files). These were combined into fastq format using convertFastaAndQualToFastq.jar script downloaded from: http://www.cbcb.umd.edu/software/PBcR/data/convertFastaAndQualToFastq.jar.

3.3 Assembly pipelines

LQS pipeline: Pipeline developed and published by Loman *et al.* in their pivotal nanopore assembly paper (https://github.com/jts/nanopore-paper-analysis). The pipeline consists of Nanocorrect, WGS and Nanopolish. The version of the pipeline tested in this paper uses Nanocorrect commit 47dcd7f147c and WGS version 8.2. For the base version of the pipeline, we didn't use Nanopolish.

PBcR: Implemented as a part of the WGS package (http://wgs-assembler.sourceforge.net/wiki/index.php/PBcR). In this paper version 8.3rc2 of WGS was used. Spec file defining assembly parameters for nanopore data, was downloaded from the PBcR web page.

FALCON: To evaluate Falcon we used the FALCON-integrate project (https://github.com/PacificBiosciences/FALCON-integrate) (commit: 3e7dd7db190). Since no formal parameter specification for nanopore data currently exists, we derived a suitable set of parameters through trial and error (Supplementary Note 1).

SPAdes: SPAdes v3.6.1 was downloaded from http://bioinf.spbau.ru/en/content/spades-download-0.

Table 1 Description of the benchmarking datasets used for evaluation

Dataset 0	Illumina reads used by hybrid assemblers, consists of three libraries: 1. frag library—paired-end reads (read length 101 bp, insert size 180 bp), coverage 55×, 1 186 191 × 2 reads, 2. jump libraries—mate-pair reads (read length 93 bp, insert size 3000 bp), total coverage 85×, 1 977 903 × 2 reads
Dataset 1	Complete *E. coli* R7.3 dataset, contains both 1d and 2d reads (both pass and fail), total coverage 67 × (70 531 reads), of which 2d reads comprise 14 × (11 823 reads)
Dataset 2	Reads from (Loman et al., 2015) subsampled to coverage 19×, pass 2d reads only (Loman et al., 2015) (in total 16 945 reads)
Dataset 3	Complete dataset used by (Loman et al., 2015) nanopore assembly paper, contains pass 2d reads only, coverage 29×, 22 270 reads
Dataset 4	Reads from MARC WTCHG dataset, 2d reads extracted from pass (33×) and fail (7×) folders, total coverage 40×, total number of 2d reads: 29 635
Dataset 5	2d reads extracted from the first run of the MAP006 dataset (MAP006-1), from pass folder only, coverage 54×, 25 483 reads in total

ALLPATHS-LG: ALLPATHS-LG release 52488 was downloaded from https://www.broadinstitute.org/software/allpaths-lg/blog/?page_id=12.

Canu: Canu was obtained from https://github.com/marbl/canu (commit: 70e711a382f).

Miniasm: Miniasm was obtained from https://github.com/lh3/miniasm (commit: 17d5bd12290). For calculating read overlaps we used Minimap (https://github.com/lh3/minimap) (commit: 1cd6ae3bc7c).

3.4 Evaluating the results

All assembly results were compared to the *E. coli* K-12 MG1655 NCBI reference, NC_000913.3. Assembly quality was evaluated using Quast 3.1 (Gurevich *et al.*, 2013) and Dnadiff (Kurtz *et al.*, 2004) tools. CPU and memory consumption was evaluated using a fork of the Cgmemtime tool (https://github.com/isovic/cgmemtime.git). For assemblies that produced one 'big contig', over 4Mpb in length, that contig was extracted and solely compared to the reference using Dnadiff tool.

Of all tested assembly pipelines, only LQS pipeline has a polishing phase (Nanopolish). To make the benchmark fair and since other assemblers' results could also be polished to further improve them, all draft assemblies containing a 'big contig' were polished using Nanopolish (https://github.com/jts/nanopolish, commit b09e93772ab4), regardless of the assembly pipeline they were generated with. At the time of testing, Nanopolish did not support 1d reads and had trouble with very small contigs. Therefore, we applied Nanopolish only to the largest contig in each assembly and skipped Dataset 1 which contains 1d reads. This does not present a problem because Dataset 1 was not successfully assembled by any of the non-hybrid assemblers.

3.5 Benchmarking framework

We developed a benchmarking framework to easily evaluate the performance of assembly tools on nanopore (or other) data. The framework is available on GitHub at https://github.com/kkrizanovic/NanoMark. Running the framework will download and install all required software and dependencies, enable assembly of a given dataset using one or more assemblers and provide evaluation of the results. The framework currently implements assembly pipelines described in this paper, but can easily be expanded to others. The framework is described in more detail on the GitHub page.

4 Results

4.1 Non-hybrid assembly quality

Since Nanopolish currently does not support 1d reads, and none of the other assemblers include a polishing phase, initially we focused on comparison of non-polished draft assemblies.

Table 2 displays assembly results on Datasets 2–5 assessed using Quast (We have observed that for lower identity sequences Genome fraction, calculated by Quast, has unexpectedly low values, while for higher identity sequences seems to be a good measure of sequence quality.) and Dnadiff. Dataset 1 analysis is omitted because of its particular characteristics. It has a greater total coverage but much lower data quality compared to other datasets (older version of the sequencing protocol, low percentage of 2d reads and the use of 1d reads). None of the non-hybrid assemblers managed to produce a good assembly using Dataset 1 (Supplementary Table 2). It can be concluded that low 2d coverage together with high coverage of low quality 1d reads is not sufficient to complete an assembly of a bacterial genome using currently available methods.

None of the non-hybrid assembly pipelines managed to complete the genome at 20× coverage. LQS pipeline produced the best assembly—it managed to cover almost the whole genome, albeit using 8 separate contigs. 30× seems to be sufficient for LQS pipeline to get very good results and for Canu and PBcR to cover most of the genome, however, with the largest contig notably shorter than the reference genome (especially for PBcR). At coverages over 40×, all tested assemblers produce good contiguous assemblies, which surprisingly includes Falcon, originally designed for PacBio data. To further estimate the coverage required by each assembler for a contiguous assembly, we used Dataset 5 subsampled to coverages 32×, 35×, 37× and 40×. The results are shown in Supplementary Table 3.

Another surprising result that can be seen in Table 2 is a noticeable drop in assembly quality for LQS pipeline on Dataset 5. While it managed to cover a greater part of the reference than any other pipeline on any dataset, its assembly consisted of 5 contigs, the largest of which is just over 4 Mbp. Overall, LQS assemblies demonstrate the highest average identity compared to the reference sequence, even without applying the polishing phase.

Miniasm's strengths are, on the other hand, oriented towards very fast and contiguous assembly of the genome, without increasing the per-base quality of the resulting sequences. Since it does not include an error correction step nor a consensus step, the contigs it produces contain errors similar to the input read data. This makes Miniasm hard to numerically compare to other assemblers without polishing the contigs. Table 2 shows that Miniasm manages to produce assemblies which contain, on average, a smaller number of contigs compared to other methods, cumulatively covering the entire genome. On coverages above 40×, Miniasm produces a single contig assembly of the entire *E. coli* genome. Since statistics in Table 2 make Miniasm hard to compare to other assemblers, we generated dotplots of the largest contigs produced for Datasets 3–5 (Supplementary Fig. 1). This was performed in order to validate that the generated contigs were not chimeric or misassembled.

Additionally, we performed a 'big contig' analysis where only the largest contig of length ≥4 Mbp (a representative of the *E. coli*

Table 2 Assembly quality assessment using Quast and Dnadiff

Data-set	Assembler	# ctg.	N50	Genome fraction (%)	Avg. Identity 1-to-1	Total SNPs	Total Indels
2	LQS	8	1159703	**99.895**	98.08	8858	79 746
	Falcon	98	11083	6.994	94.58	3263	47 211
	PBcR	22	246681	0.593	93.7	14 823	269 053
	Canu	26	332535	90.123	95.71	5691	183 774
	Miniasm	15	353994	0.002	84.21	24 8575	373 190
3	LQS	3	**4603990**	**99.998**	**98.49**	4568	65 283
	Falcon	124	13838	17.316	94.97	3206	59 638
	PBcR	1	4329903	12.825	94.03	7209	262 357
	Canu	10	4465231	88.655	95.77	5213	185 027
	Miniasm	3	3362269	0.002	84.04	24 7849	367 927
4	LQS	8	**4622531**	**99.938**	**99.08**	2256	40 118
	Falcon	13	4538244	99.938	97.66	3710	104 165
	PBcR	3	3615068	99.553	97.39	2394	11 7397
	Canu	2	4576679	99.915	98.42	812	71 878
	Miniasm	1	4577227	0.002	88.31	18 5194	326 066
5	LQS	5	4006324	**99.991**	99.43	1435	25 106
	Falcon	1	4580230	99.655	98.84	2589	50 662
	PBcR	1	4596475	99.914	98.99	1136	45 542
	Canu	1	**4600945**	**99.746**	**99.29**	415	32 100
	Miniasm	1	4774395	0.002	88.69	175 279	328 688

Bold values present the best scoring result for a particular dataset.

chromosome) was selected and evaluated using Dnadiff. This analysis gave a good estimate on the quality of the assembly from the aspects of chromosome completeness and breakage. Apart from Miniasm, all non-hybrid assemblies that produced a 'big contig' had a comparable number of breakpoints (10–50) with the exception of PBcR on Dataset 3 (841) and LQS on Dataset 5 (88). It is interesting to note that in these cases the 'big contig' is considerably shorter than the reference (see Supplementary Table 4).

Since the results for non-hybrid assembly tools show variation in assembly quality across datasets (Table 2), we further investigated the differences between them. As described in the Background section, there are two major differences: (**I**) LQS and PBcR both employ WGS (Celera) as their middle-layer assembler and Canu is a modified fork of Celera, while Falcon and Miniasm implement their own string graph layout modules; and (**II**) each of these pipelines, save for Miniasm, implements its own error–correction module. Taking into account that both Celera and Falcon utilize an overlap-graph based layout step, we suspected that (**II**) may have played a more significant role on the assembly contiguity. The error–correction process is performed very early in each pipeline, and the quality of corrected reads can directly influence any downstream analysis. For this purpose, we analyzed the error rates in raw reads from Dataset 3 (Supplementary Fig. 2) as well as the error-corrected reads generated by Nanocorrect, PBcR, Canu and Falcon error–correction modules (Supplementary Fig. 3). For analysis, all reads were aligned to the *E. coli* K-12 reference (NC_000913.3) using GraphMap (parameters '-a anchorgotoh'). The results show that each error–correction module produces corrected reads with a significantly different error profile. The raw dataset (coverage 28.78×) contained a mixture of ∼3% insertions, ∼4% deletions and ∼9% mismatches (median values). While the insertion errors were mostly eliminated by all error-correctors, PBcR and Falcon exhibited higher amounts of deletion errors in their output. Nanocorrect produced the best results, reducing both deletion and mismatch rates to 1%, while still maintaining a large coverage of the output error-corrected reads (25.85×). The error profile of Canu-corrected reads (Supplementary Fig. 3c) resembles the one obtained with Falcon error correction (Supplementary Fig. 3d). This is expected considering that Canu directly borrows components from the Falcon error–correction module.

To assess the influence of (**I**), we used error-corrected reads generated by Nanocorrect as the input data for Falcon and Miniasm for every dataset. We noticed that this procedure increased both the contiguity of Falcon's assembly and the average identity on all datasets (Supplementary Table 5). Increase in coverage of error-corrected reads provided a consistent increase of the quality of assembly in terms of largest contig length, average identity and number of variants. Although draft assemblies produced by the LQS pipeline exhibited a reduction in the size of the largest contig on Dataset 5, these assemblies also resulted in lower number of variants (SNPs and indels) compared to the Nanocorrect + Falcon combination. Miniasm benefitted from error-corrected reads as well, however, the difference in results is not as dramatic as for Falcon (Supplementary Table 6). Although the number of contigs for Datasets 1 and 2 increased when error-corrected reads were used, the total length of the generated contigs increased as well. Single contig full-genome assembly was achieved even on Dataset 3. The average identity of Nanocorrect + Miniasm assemblies is much higher than for Miniasm alone (and comparable to error-corrected datasets), which is expected as corrected reads are directly used to construct the contigs.

4.2 Hybrid pipeline comparison

Hybrid and non-hybrid assembly pipelines are not directly comparable because hybrid pipelines have an advantage in greater coverage supplied by Illumina reads. Table 3 gives a more detailed comparison between two hybrid assemblers ALLPATHS-LG and SPAdes. Besides running both pipelines on Dataset 0 (paired-end and mate-pair reads) together with each nanopore dataset, SPAdes was also tested using only Illumina paired-end reads (without mate-pair reads).

The table shows that ALLPATHS-LG produces better results than SPAdes on all datasets, from Dataset 0 without nanopore data for which SPAdes is not able to produce one sufficiently large contig,

Table 3 Comparing ALLPATHS-LG and SPAdes results

Data-set	Assembler	# ctg.	N50	Genome fraction (%)	Avg. Identity 1-to-1	Total SNPs	Total Indels
0	ALLPATHS-LG	3	**4626283**	**99.219**	**99.99**	59	73
	SPAdes PE+MP	106	1105151	99.089	99.98	231	78
2	ALLPATHS-LG	2	**4639001**	**99.938**	**99.99**	**10**	**12**
	SPAdes PE only	20	4470699	99.908	99.99	430	93
	SPAdes PE+MP	18	4488904	99.912	99.98	427	110
3	ALLPATHS-LG	3	**4638937**	**99.938**	**99.99**	**6**	**38**
	SPAdes PE only	19	4474624	99.908	99.99	418	92
	SPAdes PE+MP	19	4474608	99.88	99.99	425	108
4	ALLPATHS-LG	1	**4638952**	**99.938**	**99.99**	**8**	**20**
	SPAdes PE only	20	4475777	99.908	99.99	401	66
	SPAdes PE+MP	20	4475770	99.88	99.99	399	73
5	ALLPATHS-LG	1	**4638958**	**99.938**	**99.99**	**3**	**5**
	SPAdes PE only	18	**4648869**	**99.918**	**99.99**	421	47
	SPAdes PE+MP	16	**4648863**	**99.918**	**99.99**	420	53

Bold values present the best scoring result for a particular dataset.

to Dataset 5 on which the difference is miniscule and apparent only in the number of SNPs and indels.

It is interesting to note that while ALLPATHS-LG requires both a paired-end and a mate-pair library to run, SPAdes seems not to be able to leverage mate-pair reads to a noticeable effect. In the presence of nanopore reads, results using paired-end Illumina library without mate-pairs seems to be equal to or even slightly better than with a mate-pair library, for all nanopore datasets. This means that in a situation where mate-pair reads are unavailable, SPAdes might be a good choice for a *de novo* assembler.

While none of the non-hybrid assemblers managed to produce a good assembly using Dataset 1 (Supplementary Table 2), both hybrid assemblers were able to use it to improve their assembly. From Tables 2 and 3, it can be concluded that hybrid assembly pipelines achieve better results than non-hybrid ones. However, this is mostly because Illumina reads provide additional coverage of the genome.

4.3 Resource usage

To estimate efficiency of each assembly pipeline, NanoMark measures and reports User time, System time, CPU time, Real time (Wall clock time) and Maximum memory usage (Resident Set Size, RSS) for each assembly tool and dataset.

Table 4 shows CPU time and memory measurements with respect to sequencing coverage for each assembly pipeline. Miniasm proved to be by far the fastest of the tested assemblers, while LQS was, also by a large margin, the most time consuming. We note that, in general, error–correction of non-hybrid assemblers is the most time consuming step, which is especially evident in the LQS pipeline. Miniasm completely skips error–correction and consensus steps and is approximately three orders of magnitude faster than the second fastest non-hybrid tool (Canu) and two orders of magnitude faster than the fastest tested hybrid tool (SPAdes). All assembly pipelines consumed less than 20 GB of memory, with Miniasm being the most conservative one.

4.4 Polishing the assembly

For every assembly result that contained a contig at least 4 Mbp in length, we extracted that contig, and polished it using Nanopolish. The results were then compared to the reference using Quast and Dnadiff. The results of the analysis are shown in Table 5.

We can see that, without an exception, Nanopolish will improve a non-hybrid assembly. Contig length (N50) will come closer to the reference length, average identity will increase while total number of SNPs and indels will decrease. On the other hand, the effect on hybrid assemblies is opposite. Contig length (N50) will usually decrease, average identity will always decrease, while total number of SNPs and indels will increase.

Although surprising at first, this result is expected if we consider that with hybrid assemblies Nanopolish is trying to improve contigs obtained from data with lower error rate (Illumina reads) using data with higher error rate (nanopore 2d reads).

Table 4 CPU time (hours)/Maximum memory usage (GB)

	Coverage			
Assembler	20	30	40	50
LQS	1086 / 4	2539 / 4	4438 / 4	8142 / 4
ALLPATHS-LG	13.2 / 18	26.0 / 18	44.9 / 17	144.5 / 20
PBcR	6.2 / 2	13.7 / 6	14.1 / 5	19.3 / 5
Falcon	3.1 / 6	6.4 / 10	19.7 / 10	13.8 / 13
SPAdes	0.9 / 5	1.0 / 5	1.1 / 5	1.2 / 5
Canu	5.33 / 3	11.2 / 4	28.7 / 4	9.39 / 4
Miniasm	0.009 / 2	0.015 / 2	0.026 / 3	0.044 / 4

5 Conclusion

In our study we developed a benchmarking framework for *de novo* assembly tools focused on third generation sequencing data and compared several hybrid and non-hybrid *de novo* assemblers as well as assessed their ability to work with nanopore sequencing data. Each examined tool proved capable of assembling a whole bacterial genome under the right conditions. Needless to say, the choice of the best assembly tool will heavily depend upon the characteristics of the dataset. Keeping in mind that hybrid and non-hybrid assemblers are not directly comparable, we can say that ALLPATHS-LG showed overall the best results. However, it requires a rather specific set of Illumina paired-end and mate-pair short read libraries to perform the assembly, which might not always be practical to obtain. In case only paired-end reads are available, SPAdes might be a good choice. Of the non-hybrid assembly tools, on some datasets LQS pipeline came close to or even surpassed hybrid tools. However, extremely high CPU time used by Nanocorrect might make it

Table 5 Polished assembly quality assement

Dataset	Assembler	Polish	N50	Genome fraction (%)	Avg. Identity 1-to-1	Total SNPs	Total Indels
2	Allpaths	YES	4638854	99.937	99.46	1012	24 213
	Allpaths	NO	4639001	99.938	99.99	10	12
	SPAdes	YES	4489211	96.227	99.44	1011	23 463
	SPAdes	NO	4488904	96.220	99.98	424	110
3	LQS	YES	4648870	99.997	99.47	1357	22 622
	LQS	NO	4603990	99.998	98.49	4568	65 283
	PBcR	YES	4615494	99.458	99.22	4460	31 520
	PBcR	NO	4329903	12.825	94.03	7209	262 357
	Canu	YES	4638184	99.783	99.28	3180	29 607
	Canu	NO	4465231	88.644	95.77	5318	186 843
	Allpaths	YES	4636408	99.936	99.53	790	21 010
	Allpaths	NO	4638937	99.938	99.99	6	38
	SPAdes	YES	4472428	96.226	99.52	769	20 291
	SPAdes	NO	4474608	96.220	99.99	424	108
4	LQS	YES	4664571	99.938	99.60	890	17 712
	LQS	NO	4622531	99.938	99.08	2256	40 118
	Falcon	YES	4643699	99.937	99.54	1829	19 361
	Falcon	NO	4538244	99.938	97.66	3710	104 165
	Canu	YES	4653892	99.934	99.58	1196	18 012
	Canu	NO	4576679	99.915	98.42	812	71 878
	Miniasm	YES	4667580	99.898	98.3	21 331	58 211
	Miniasm	NO	4577227	0.002	88.31	185 194	326 000
	Allpaths	YES	4647282	99.938	99.62	674	17 168
	Allpaths	NO	4638952	99.938	99.99	8	20
	SPAdes	YES	4484185	96.223	99.61	655	16 573
	SPAdes	NO	4475770	96.220	99.99	398	73
5	LQS	YES	4018309	86.570	99.80	453	7671
	LQS	NO	4006324	86.570	99.43	1237	21 852
	Falcon	YES	4624811	99.654	99.78	834	9318
	Falcon	NO	4580230	99.655	98.84	2589	50 662
	PBcR	YES	4639491	99.973	99.79	627	9131
	PBcR	NO	4596475	99.914	98.99	1136	45 542
	Canu	YES	4631443	99.918	99.8	556	8547
	Canu	NO	4600945	99.786	99.29	415	32 100
	Miniasm	YES	4696482	99.712	98.06	20 395	70 406
	Miniasm	NO	4774395	0.002	88.69	175 279	328 688
	Allpaths	YES	4637979	99.938	99.82	312	7859
	Allpaths	NO	4638958	99.938	99.99	3	5
	SPAdes	YES	4648653	99.929	99.82	343	7904
	SPAdes	NO	4648863	99.918	99.99	420	53

prohibitively slow on larger genomes and larger datasets, in which case Canu, Falcon or PBcR could be used instead.

Miniasm must be considered apart from other assemblers. While the lack of error correction and consensus phases results in assemblies with a significant error rate, astonishing CPU efficiency makes it a perfect candidate for time-critical applications, especially if it could be enhanced with a suitably efficient consensus phase. Applying Nanopolish to Miniasm assemblies showed that they could be significantly improved, but at this point still fall behind other assemblies.

Polishing draft assemblies with Nanopolish improved the results of non-hybrid assemblies, but worsened the results of hybrid ones. Relative assembly quality remained the same, but after polishing the difference between hybrid and non-hybrid assemblies reduced substantially.

The high contiguity of nanopore-only assemblies provides a number of other important opportunities. For example, even small bioinformatics laboratories can now study genomic structural variations and rearrangements, or identify large antibiotic resistance islands in genomes, for which exact base variations are not of such high importance; all in-house.

We can expect that with further development of nanopore technology (and other long read sequencing technologies) read quality will increase and the technology will become more accessible and more affordable. This will make *de novo* assembly using nanopore reads faster, more precise and applicable to larger genomes.

Funding

This work has been supported in part by Croatian Science Foundation under the project UIP-11-2013-7353 'Algorithms for Genome Sequence Analysis'. I.S. is supported in part 25 by the Croatian Academy of Sciences and Arts under the project 'Methods for alignment and assembly of DNA sequences using nanopore sequencing data'.

Conflict of Interest: none declared.

References

Bankevich,A. *et al*. (2012) SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J. Comput. Biol.*, **19**, 455–477.

Bankevich,A. and Pevzner,P.A. (2016) TruSPAdes: barcode assembly of TruSeq synthetic long reads. *Nat. Methods*, **13**, 248–250.

Berlin,K. *et al.* (2015) Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nat. Biotechnol.*, **33**, 623–630.

Chaisson,M.J. and Tesler,G. (2012) Mapping single molecule sequencing reads using Basic Local Alignment with Successive Refinement (BLASR): Theory and Application. *BMC Bioinformatics*, **13**, 238.

Chin,C.S. *et al.* (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat. Methods*, **10**, 563–569.

Gnerre,S. *et al.* (2011) High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci. USA*, **108**, 1513–1518.

Goldberg,S.M.D. *et al.* (2006) A Sanger/pyrosequencing hybrid approach for the generation of high-quality draft assemblies of marine microbial genomes. *Proc. Natl. Acad. Sci. USA*, **103**, 11240–11245.

Gurevich,A. *et al.* (2013) QUAST: quality assessment tool for genome assemblies. *Bioinformatics*, **29**, 1072–1075.

Ip,C.L.C. *et al.* (2015) MinION Analysis and Reference Consortium: Phase 1 data release and analysis. *F1000Research*, **4**, 1075.

Jain,M. *et al.* (2015) Improved data analysis for the MinION nanopore sequencer. *Nat. Methods*, **12**, 351–356.

Koren,S. *et al.* (2012) Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat. Biotechnol.*, **30**, 693–700.

Kurtz,S. *et al.* (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12.

Laver,T. *et al.* (2015) Assessing the performance of the Oxford Nanopore Technologies MinION. *Biomol. Detect. Quantif.*, **3**, 1–8.

Li,H. (2013) Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997v1 [q-bio.GN].

Li,H. (2016) Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics, doi: 10.1093/bioinformatics/btw152.

Liao,Y.C. *et al.* (2015) Completing bacterial genome assemblies: strategy and performance comparisons. *Sci. Rep.*, **5**, 8747.

Loman,N.J. *et al.* (2015) A complete bacterial genome assembled de novo using only nanopore sequencing data. *Nat. Methods*, **12**, 733–735.

Loman,N.J. and Quinlan,A.R. (2014) Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, **30**, 3399–3401.

Miller,J.R. *et al.* (2008) Aggressive assembly of pyrosequencing reads with mates. *Bioinformatics*, **24**, 2818.

Myers,E.W. *et al.* (2000) A whole-genome assembly of Drosophila. *Science*, **287**, 2196–2204.,

Myers,E.W. (2005) The fragment assembly string graph. **21**, 79–85.

Myers,G. (2014) Efficient Local Alignment Discovery amongst Noisy Long Reads. In: Brown,D. and Morgenstern,B. (eds) *Algorithms in Bioinformatics, Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 52–67.

Nagarajan,N. and Pop,M. (2013) Sequence assembly demystified. *Nat. Rev. Genet.*, **14**, 157–167.

Pop,M. (2009) Genome assembly reborn: recent computational challenges. *Brief. Bioinform.*, **10**, 354–366.

Quick,J. *et al.* (2014) A reference bacterial genome dataset generated on the MinION(TM) portable single-molecule nanopore sequencer. *Gigascience*, **3**, 22.

Schirmer,M. *et al.* (2015) Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. *Nucleic Acids Res.*, **43**, e37.

Sović,I. *et al.* (2016) Fast and sensitive mapping of error-prone nanopore sequencing reads with GraphMap. *Nat. Commun.*, **7**, 11307.