BRIEF COMMUNICATION
https://doi.org/10.1038/s43588-021-00073-4

nature computational science

Time- and memory-efficient genome assembly with Raven

Robert Vaser[1,2] and Mile Šikić[1,2] ✉

Whole genome sequencing technologies are unable to invariably read DNA molecules intact, a shortcoming that assemblers try to resolve by stitching the obtained fragments back together. Here, we present methods for the improvement of de novo genome assembly from erroneous long reads incorporated into a tool called Raven. Raven maintains similar performance for various genomes and has accuracy on par with other assemblers that support third-generation sequencing data. It is one of the fastest options while having the lowest memory consumption on the majority of benchmarked datasets.

Sequencing technologies have come a long way, from sequencing tiny fragments, in their infancy, to the large chunks obtainable today. The relentless advances in the length and accuracy that are achievable continue to alleviate the puzzle-like reconstruction problem of the sequenced genome, as more repetitive structures can be resolved naturally. Amid the excess of available state-of-the-art options for de novo genome assembly[1–6], we present a fast, memory-frugal, reliable and easy-to-use tool, called Raven. It is an overlap–layout–consensus assembler that accelerates the overlap step, builds an assembly graph[4] from reads that were pre-processed with pile-o-grams[7], implements a simplification method based on graph drawings, and polishes the unambiguous graph paths with Racon[8], all compiled into a single executable.

Short substring matching is a conventional approach for similarity searches in bioinformatics[9,10]. However, even with minimizers[4], the overlap step of de novo assembly can take a substantial amount of time when handling larger genomes. To tackle this problem, we enhanced the minimap[4] algorithm following the MinHash approach[11], where we select a fixed number of lexicographically smallest minimizers as the sequence sketch. The combination of MinHash on top of minimizers has already been explored within the sequence mapper MashMap[12], while a similar idea with hierarchical minimizers is the core of the de novo assembler Peregrine[13]. Based on empirical evaluations, we opted for retaining $|read|/k$ minimizers per read, where k is the minimizer length. Without any other algorithmic modifications to minimap, we are able to identify contained reads and create pile-o-grams for read pre-processing in a fraction of time and with a small impact on sensitivity. Suffix–prefix overlaps needed for graph constructions are found with the unmodified minimap algorithm within the containment-free read set, which is usually smaller than the whole sequencing yield by almost an order of magnitude.

Raven loads the whole sequencing sample into memory in compressed form, and finds overlaps in fixed-size blocks to decrease the memory footprint. Found overlaps are immediately transformed into pile-o-grams and discarded, except the longest few per read, which are used for containment removal. Chimeric reads are iteratively identified and chopped by detecting sharp declines of coverage in pile-o-grams using coverage medians inferred from the stored overlaps. As minimap ignores the most frequent minimizers, which are critical for good repeat annotations, we lower this threshold while overlapping all contained reads to the set of containment-free reads, and search the updated pile-o-grams for sharp coverage inclines followed by sharp declines, both above the coverage median. Afterwards, the containment-free read set is overlapped to itself, and repeat annotations are used to remove false overlaps between reads containing repetitive regions. Once the assembly graph is created, it is simplified stepwise with transitive reduction, tip removal and bubble popping. Eventually, we simplify the graph with a method that lays out the graph in a two-dimensional (2D) Euclidean space, searches for edges that connect distant parts of the graph, then removes them. Applying the force-directed placement algorithm[14], which draws tightly connected vertices together, we can distinguish undetected chimeric or repeat-induced edges that are elongated with respect to others due to their rareness (Fig. 1). Collapsing unambiguous paths while leaving room near junction vertices, coupled with the hierarchical force-calculation algorithm[15], makes this drawing-based simplification method feasible for even the largest assembly graphs. To finalize the assembly, contiguous paths of the graph are passed to two rounds of Racon.

Given that an earlier version of Raven proved to be one of the best performers in a comprehensive benchmarking study at the prokaryotic level[16], we evaluated several state-of-the-art assemblers alongside Raven on three human datasets obtained with third-generation sequencing technologies (Table 1)—Pacific Biosciences (PacBio) and Oxford Nanopore Technologies (ONT). Owing to the recent emergence of PacBio's High-Fidelity (HiFi) sequencing protocol, which substantially improves read accuracy compared to older continuous long-read (CLR) protocols, we also evaluated one of the assemblers[13,17,18] suitable for this data type. Including default assembly quality metrics such as contiguity, genome fraction and accuracy, we evaluated gene completeness and the number of bacterial artificial chromosomes (BACs) resolved in an assembly, where possible. Details of the computational cost of each assembler are provided in Supplementary Table 1. An evaluation on older sequencing datasets is presented in Supplementary Table 2, while Raven's performance on plant genomes from two scientific studies[19,20] is described in Supplementary Table 3.

On erroneous data, Raven is one of the fastest assemblers and uses the least amount of memory on all but two datasets, while having better or comparable contiguity and accuracy. It particularly stands out in the number of contigs with similar genome reconstruction fractions, as well as in the number of retained multi-copy genes and resolved BACs on human datasets. On the other hand, Raven does not utilize the accuracy of HiFi reads, which results in longer running times and subpar assembly results on more

[1] Laboratory for Bioinformatics and Computational Biology, Faculty of Electrical Engineering and Computing, University of Zagreb, Zagreb, Croatia.
[2] Laboratory of AI in Genomics, Genome Institute of Singapore, A*STAR, Singapore, Singapore. ✉e-mail: mile_sikic@gis.a-star.edu.sg

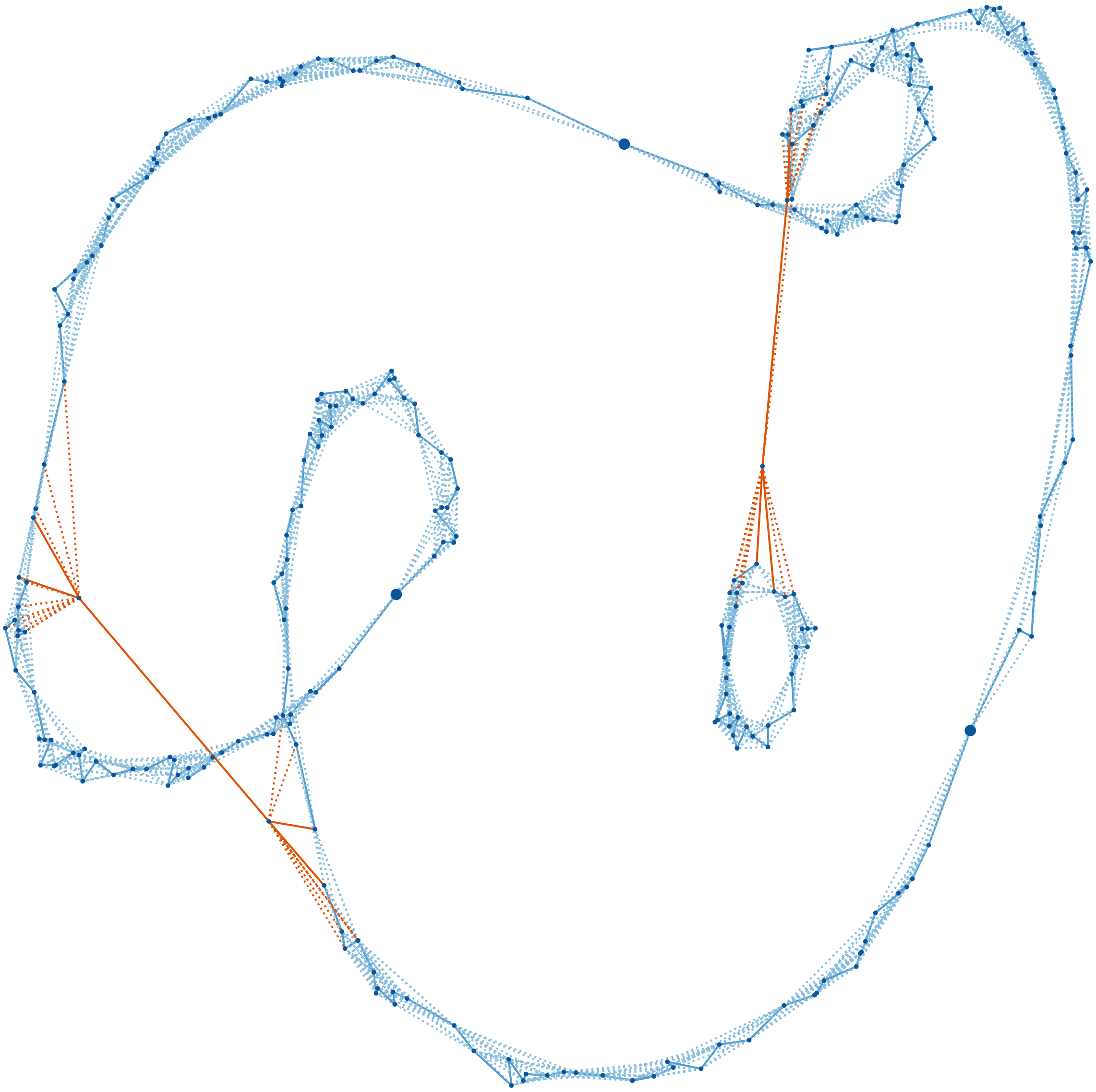

Fig. 1 | Bacterial assembly graph drawn with the force-directed placement algorithm. Raven uses vertex distances in a 2D Euclidean space to find elongated edges (orange) that connect junction vertices and removes the longest ones. Those represent false connections that occur either due to sequencing errors or repetitive genomic regions. Without unitig creation (large blue filled circles) and the hierarchical force calculation, the drawing algorithm would take an extensive amount of time on larger genomes. Transitive edges (dotted blue lines) are reinstated to increase the connectivity of neighboring vertices.

accurate data. We believe that more carefully tweaked parameters for the overlap step will lead to performance improvements. On plant datasets for *Brassica oleracea*, *Brassica rapa* and *Musa schizocarpa*, Raven produces assemblies comparable to those obtained with Ra[21]. Furthermore, both *Oryza sativa* assemblies are more contiguous than those reported with Flye, but the BUSCO[22] scores are lower as we did not polish our assemblies with Illumina data.

The presented results indicate that PacBio HiFi assemblers achieve better overall reconstruction metrics, although ONT assemblies do not fall far off. ONT sequencing is still more approachable because of the affordable consumables and portable devices it needs, while requiring less genomic DNA than regular PacBio protocols. In addition, the length advantage of ONT reads and the recent increase in accuracy with the newest version of the Bonito basecaller (still in the testing phase) justify the use of assemblers that support this technology. We argue that Raven's performance, coupled with the reduced cost per base of long-read sequencing technologies, will enable the high-quality assembly of large genomes, even for laboratories with limited funding.

Methods

The Raven workflow. The algorithm begins by constructing pile-o-grams (1D structures storing per-base coverage) and removing contained reads with the minimap algorithm, using 15-mers, a sliding window of five bases and discarding 10^{-3} most frequent minimizers. The whole sequencing dataset is loaded into memory, replacing nucleotides with two bits, and merging 64 succeeding Phred quality scores with their average. Reads are overlapped to each other in chunks of 1 Gbp versus 4 Gbp, and only the lexicographically smallest $|read|/15$ minimizers are picked in both the index and the query (details are provided in Supplementary Figs. 1–3 and an accuracy comparison in Supplementary Table 4). Once a block is processed, all overlaps are stacked into pile-o-grams, which are decimated to every 16th base. The longest 16 overlaps per read are stored for containment removal and connected component retrieval. When all pairwise overlaps are obtained, coverage medians are calculated for each pile-o-gram, reads are trimmed to the longest region covered with at least four other reads, and potential chimeric sites are detected by finding bases that have 1.82 times smaller coverage than their neighboring bases. Contained reads are dropped only if the contained read does

Table 1 | Evaluation of long-read assemblers across sequencing technologies

Dataset	Metric	ONT					PacBio CLR				PacBio HiFi	
		Raven	Canu	Flye	Shasta	Wtdbg2	Raven	Flye	Shasta	Wtdbg2	Raven	hifiasm
H. sapiens CHM13 ONT ~130× PacBio CLR ~50× PacBio HiFi ~35×	Genome fraction (%)	93.39	**94.94**	93.44	92.55	88.67	91.83	**92.12**	91.44	91.78	92.55	99.78
	No. of contigs	**120**	558	548	1,236	19,029	**897**	2,247	2,937	3,632	1,755	470
	NG50 (Mb)	67.6	**79.5**	68.4	41.1	5.3	11.0	**20.8**	12.7	16.8	12.0	88.9
	NGA50 (Mb)	56.6	44.7	**56.8**	28.9	2.3	9.4	**17.5**	11.6	14.6	10.4	80.8
	NGA75 (Mb)	32.1	19.8	**32.2**	12.0	0.6	3.8	**6.0**	3.7	4.2	3.7	36.4
	No. of misassemblies	2,847	3,885	264	**126**	7,046	869	316	**186**	954	2,921	156
	Mismatch fraction (%)	0.07	0.12	**0.01**	0.04	0.28	0.04	**0.02**	0.03	0.07	0.06	0.00
	Indel fraction (%)	**0.09**	0.48	**0.09**	0.35	0.43	0.09	**0.02**	0.25	0.13	0.01	0.00
	Quality value	32.61	24.33	**32.64**	25.70	27.01	32.03	**39.95**	27.04	32.30	43.51	52.69
	Single-copy genes (%)	98.94	93.59	**99.28**	95.82	82.98	98.42	**98.47**	96.57	96.68	98.29	99.91
	Duplicated genes (%)	0.33	0.16	0.15	**0.01**	5.29	0.30	0.25	**0.02**	0.06	0.39	0.06
	Multi-copy genes (%)	**86.22**	49.51	62.55	14.61	34.83	**44.42**	30.26	5.39	6.59	44.64	99.70
	Resolved BACs (%)	**95.52**	88.72	72.80	43.89	30.29	**42.04**	36.79	33.08	35.86	39.10	96.60
	CPU time (h)	**4,792**		4,855		5,978	498	1,865	**36**	461	554	
	Memory (GB)	**251**		873		423	**98**	407	547	180	65	
H. sapiens HG002 ONT ~60× PacBio CLR ~80× PacBio HiFi ~35×	Genome fraction (%)	92.69	**94.03**	93.31	93.45	88.87	91.26	**91.71**	89.47	90.88	92.14	96.18
	No. of contigs	**192**	767	776	2,039	10,166	**2,168**	2,879	8,425	4,660	2,375	383
	NG50 (Mb)	34.5	32.6	**50.4**	28.9	7.7	3.6	**11.6**	0.9	9.9	6.5	98.2
	NGA50 (Mb)	21.1	20.1	**26.8**	22.7	3.4	2.9	**9.0**	0.9	7.6	5.9	31.4
	NGA75 (Mb)	9.7	8.0	**12.6**	11.4	1.2	1.2	**3.1**	0.3	2.3	2.1	13.1
	Mismatch fraction (%)	0.16	0.22	**0.14**	0.15	0.38	0.14	**0.13**	0.14	0.19	0.18	0.24
	Indel fraction (%)	0.23	0.79	0.22	**0.18**	0.61	0.16	**0.05**	0.36	0.20	0.04	0.03
	Quality value	28.03	21.89	28.22	**29.18**	24.31	29.46	**37.42**	25.65	29.46	42.27	48.68
	Single-copy genes (%)	97.83	88.95	98.23	**98.52**	85.74	96.73	**97.64**	90.57	93.37	97.57	99.24
	Duplicated genes (%)	0.60	0.55	0.48	**0.15**	2.73	0.40	0.32	**0.02**	0.04	0.48	0.30
	Multi-copy genes (%)	**70.94**	28.39	56.70	27.87	15.66	**26.74**	18.65	4.04	5.17	38.73	85.24
	CPU time (h)	1,157		1,962	**128**	2,191	987	3,586	**34**	544	527	
	Memory (GB)	**105**		951	771	352	**129**	562	567	207	67	
H. sapiens HG00733 ONT ~80× PacBio CLR ~95× PacBio HiFi ~35×	Genome fraction (%)	92.51	**94.04**	92.72	92.90	89.18	**92.34**	92.33	92.07	90.80	91.96	96.09
	No. of contigs	**262**	778	1,028	1,953	4,848	**559**	1,589	2,281	2,863	2,176	657
	NG50 (Mb)	33.3	**40.6**	37.7	18.4	13.9	22.5	26.5	14.0	**29.1**	7.1	68.3
	NGA50 (Mb)	18.3	22.5	**23.9**	13.5	8.2	17.3	18.0	12.2	**19.4**	6.1	29.9
	NGA75 (Mb)	8.3	**9.5**	9.4	5.4	2.4	**7.3**	6.8	4.3	6.4	2.2	12.8

Continued

Table 1 | Evaluation of long-read assemblers across sequencing technologies (continued)

Dataset	Metric	ONT					PacBio CLR				PacBio HiFi	
		Raven	Canu	Flye	Shasta	Wtdbg2	Raven	Flye	Shasta	Wtdbg2	Raven	hifiasm
	Mismatch fraction (%)	**0.13**	0.20	**0.13**	**0.13**	0.27	0.13	**0.11**	0.17	0.17	0.16	0.22
	Indel fraction (%)	0.35	0.68	0.41	**0.21**	0.72	0.14	**0.04**	0.38	0.23	0.03	0.03
	Quality value	25.71	22.64	24.98	**27.98**	22.77	29.76	**37.31**	25.14	28.35	40.06	42.39
	Single-copy genes (%)	97.14	91.41	96.81	**97.65**	88.48	98.41	**98.67**	96.04	96.44	97.58	99.33
	Duplicated genes (%)	0.42	0.71	0.28	**0.07**	0.25	0.50	0.26	**0.04**	0.05	0.49	0.37
	Multi-copy genes (%)	**53.78**	34.31	41.80	14.46	4.72	**56.93**	36.48	5.54	11.46	37.15	88.69
	Resolved BACs (%)	**71.05**	67.89	42.63	26.84	15.79	**48.42**	34.21	22.11	29.47	22.11	80.00
	CPU time (h)	1,234		2,871	**98**	1,895	1,522	6,473	**115**	1,491	486	
	Memory (GB)	**131**		546	870	345	**138**	663	1,012	340	70	

Values in columns that are missing central processing unit (CPU) time and memory were obtained with assemblies from other publications. The NG50 metric is defined as the length of the contig, which, coupled with longer contigs, covers 50% of the reference genome. NGA50 is calculated the same way but on top of alignments between contigs and the reference. 'Quality value' denotes the Phred base error rate of the assembly obtained by comparing k-mers between short accurate reads and the assembly. Multi-copy genes are those that occur multiple times both in the reference and the assembly. 99.5% of bases of a BAC need to be present in the assembly for it to be resolved. Bold values represent the best metric scores.

not have a potential chimeric region. Decreasing the number of reads through containment removal enables faster verification of chimeric annotations. Given the stored suffix–prefix overlaps, Raven finds connected components and their coverage median, which approximates the sequencing depth. Each annotated coverage drop is used to chop problematic reads to their longest non-chimeric region, if the drop is consistent with the coverage median of the connected component to which the read belongs. The whole process is done iteratively to capture different molecule copy numbers, because resolving chimeric reads tends to the forming of new connected components. Another containment check is carried out once the chimeric sequences are resolved.

Afterwards, Raven searches for suffix–prefix overlaps between the remaining reads, enforcing the use of all minimizers. In addition, all contained reads are overlapped to the containment-free read set to increase the coverage of repetitive regions, again employing the MinHash approach. Decreasing the minimizer frequency filter to 10^{-5} enables proper repeat annotation in which sought bases need to have a coverage that is at least 1.42 times larger than the component coverage median. Repetitive regions at either end of a read are used to iteratively remove false overlaps, that is, overlaps that connect different copies of bridged repeats (repetitive genomic regions that are entirely contained in at least one read).

Once the overlap set is cleaned, the assembly graph is built and simplified stepwise with standard layout algorithms such as transitive reduction, tipping and bubble popping. Information about transitive connections is kept for the last simplification step, which plots the assembly graph in a 2D space, to increase the connections between neighboring vertices. Raven searches for edges connecting remote parts of the graph, which are usually present due to leftover sequencing artifacts or unresolved repeats. The force-directed placement algorithm enlarges most of such edges due to their rareness. Given the quadratic time complexity $O(|V|^2)$[14] and an approximate of 100 iterations until convergence, we shrink the graph by creating unitigs (paths in the graph consisting of vertices with only one ingoing and one outgoing edge) that are 42 vertices away from any junction vertex (vertices with more than one outgoing or ingoing edge). Furthermore, approximating the forces of distant vertices by replacing them with their center of mass enables linearithmic time complexity $O(|V|\log|V|)$[15] and the use of this method on larger genomes. Depending on vertex distances in a finished drawing, Raven removes outgoing edges that are at least twice as long as any other outgoing edge of that junction vertex. As the drawing heavily depends on an initial layout, which is random but with a fixed seed, the whole procedure is restarted 16 times. It should be noted that if there exist a lot of false connections in a single area of the graph (usually induced by repeats), the drawing algorithm will not be able to sufficiently enlarge all of these edges for removal (Supplementary Fig. 4).

Finally, paths of the assembly graph without external branches are polished with a library version of Racon, using small windows of 500 nucleotides and partial order alignment with linear gaps, in a total of two iterations. All constant values used in various Raven stages were empirically determined based on publicly available datasets from the NCTC 3000 database (https://www.sanger.ac.uk/resources/downloads/bacteria/nctc/) by evaluating assembly reconstruction metrics.

Assembly evaluation. Because of resource limitations, we chose the best-performing genome assemblers for erroneous third-generation data from recent scientific papers[3,6,16]. These assemblers are Raven (v1.3.0), Canu (v2.0), Flye (v2.8.1), miniasm (v0.3-r179) coupled with minimap (v0.2-r123) and polished with two iterations of Racon (v1.4.13), Ra (v0.2.1), Shasta (v0.7.0) and Wtdbg2 (v2.5). Raven was run without any additional parameters on the ONT and PacBio CLR datasets. On PacBio HiFi datasets, we increase the k-mer length from 15 to 29 and the window length from 5 to 9 to decrease the number of found pairwise overlaps (a comparison with default parameters is provided in Supplementary Table 5). We use options '-pacbio' or '-nanopore' for Canu, '-pacbio-raw' or '-nano-raw' for Flye, '-x ont' or '-x pb' for Ra, '-x sq', '-x rs' or '-x ont' for Wtdbg2, and configuration files Nanopore-Dec2019, Nanopore-Sep2020 or PacBio-CLR-Dec2019 for Shasta. For ONT runs we modified the Shasta consensus caller to better match the basecaller used to obtain the corresponding dataset, and we decreased the minimal read length to 5,000 for non-human datasets, except the PacBio CLR *Drosophila melanogaster* dataset, for which Shasta produced a decent assembly. Canu and Wtdbg2 require approximate genome sizes, which were 120 Mb, 144 Mb and 3 Gb for the *Arabidopsis thaliana*, *D. melanogaster* and *H. sapiens* datasets, respectively. All assemblers were run with 64 threads on a server with 1 TB of RAM and two AMD EPYC 7702 64-core processors. Because of the high memory requirements, the ONT CHM13 dataset was benchmarked with 48 threads on a server with two Intel Xeon Platinum 8260L 24-core processors and 1.5 TB of Optane persistent memory. Shasta was unable to assemble the PacBio CLR HG00733 dataset on the first machine due to memory requirements, so it was run on the second machine. Also, it was not able to assemble the ONT CHM13 dataset on either machine, so we found the assembly in its publication. Canu was not run on human datasets because of its long running time, but we found assemblies in other publications[5,23]. Hifiasm human assemblies were found in its publication[18].

We used QUAST-LG[24] (v5.0.2) for assembly evaluation and ran it with a minimal identity of 80%. For *H. sapiens* datasets we used the T2T (telomere-to-telomere) reconstruction of CHM13 (and options '–large' in QUAST), while for the *A. thaliana* and *D. melanogaster* datasets we used appropriate NCBI assemblies or references depending on the strain. The assembly quality value was obtained with yak (v0.1), which is available at https://github.com/lh3/yak, by comparing 31-mers found in short accurate reads and the assembly for datasets NA12878, CHM13, HG002 and HG00733. Gene completeness was evaluated with paftools (v2.17-r982) asmgene function, found inside the minimap2[25] package. We mapped annotated Ensembl cDNA sequences (v102 for *D. melanogaster* and *H. sapiens* and v49 for *A. thaliana*) to the references and the assemblies. An identity of 97% was used to find single-copy and duplicated single-copy genes, and 99% identity was used for multi-copy genes. We validated the BAC resolution with a pipeline available at https://github.com/skoren/bacValidation (commit 4f3e463). We used VMRC53 (237 BACs), VMRC59 (647 BACs) and VMRC62 (190 BACs) clones for NA12878, CHM13 and HG00733, respectively. BUSCO (v4.1.4) scores for the five plant datasets were found with the Embryophyta database, although the current version contains more orthologs (1,614 in total).

Data availability

The ONT dataset for *A. thaliana* is available under accession no. ERR2173373, for *D. melanogaster* under SRR6702603, for *H. sapiens* NA12878 at https://github.com/nanopore-wgs-consortium/NA12878 (release 6), for *H. sapiens* CHM13 at https://github.com/marbl/CHM13 (release 6), for *H. sapiens* HG002 at https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/Ultralong_OxfordNanopore/guppy-V3.4.5/ and for *H. sapiens* HG00733 at https://s3-us-west-2.amazonaws.com/human-pangenomics/index.html?prefix=NHGRI_UCSC_panel/HG00733/nanopore/. The PacBio CLR dataset for *A. thaliana* is available at https://downloads.pacbcloud.com/public/SequelData/ArabidopsisDemoData/, for *D. melanogaster* under accession no. SRR5439404, for *H. sapiens* CHM13 at https://github.com/marbl/CHM13 (extracted from draft v1.0 bam), for *H. sapiens* HG002 at https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/PacBio_MtSinai_NIST/PacBio_fasta/ and for *H. sapiens* HG0073 under SRR7615963. The PacBio HiFi dataset for *H. sapiens* CHM13 is available from accession nos. SRR11292120–SRR11292123, for *H. sapiens* HG002 under SRR10382244, SRR10382245, SRR10382248 and SRR10382249, and for *H. sapiens* HG00733 under ERX3831682. Illumina reads for yak evaluation are available from accession nos. SRX1049768–SRX1049782 for *H. sapiens* NA12878, from https://github.com/marbl/CHM13 (extracted from draft v1.0 bam) for *H. sapiens* CHM13, from https://ftp-trace.ncbi.nlm.nih.gov/giab/ftp/data/AshkenazimTrio/HG002_NA24385_son/NIST_HiSeq_HG002_Homogeneity-10953946/NHGRI_Illumina300X_AJtrio_novoalign_bams/ (extracted from 60x bam) for *H. sapiens* HG002 and under accession no. SRR7782677 for *H. sapiens* HG00733. ONT plant datasets are available under accession nos. ERR2564160–ERR2564170 for *B. rapa*, from ERR2564373–ERR2564382 for *B. oleracea*, from ERR2571286–ERR2571303 for *M. schizocarpa*, from ERR3476478–ERR3476482 for *O. sativa basmati 334* and from ERR3476463–ERR3476466 for *O. sativa dom sufid*. All generated assemblies in this research are available at Zenodo[26].

Code availability

The Raven source code is available under an MIT license on GitHub at https://github.com/lbcb-sci/raven. Source code for version 1.3.0 used in this manuscript is also available at Zenodo[27].

Received: 23 February 2021; Accepted: 20 April 2021;
Published online: 20 May 2021

References

1. Koren, S. et al. Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res.* **27**, 722–736 (2017).
2. Chin, C.-S. et al. Phased diploid genome assembly with single-molecule real-time sequencing. *Nat. Methods* **13**, 1050–1054 (2016).
3. Kolmogorov, M., Yuan, J., Lin, Y. & Pevzner, P. A. Assembly of long, error-prone reads using repeat graphs. *Nat. Biotechnol.* **37**, 540–546 (2019).
4. Li, H. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics* **32**, 2103–2110 (2016).
5. Shafin, K. et al. Nanopore sequencing and the Shasta toolkit enable efficient de novo assembly of eleven human genomes. *Nat. Biotechnol.* **38**, 1044–1053 (2020).
6. Ruan, J. & Li, H. Fast and accurate long-read assembly with wtdbg2. *Nat. Methods* **17**, 155–158 (2020).
7. Kamath, G. M., Shomorony, I., Xia, F., Courtade, T. A. & Tse, D. N. HINGE: long-read assembly achieves optimal repeat resolution. *Genome Res.* **27**, 747–756 (2017).
8. Vaser, R., Sović, I., Nagarajan, N. & Šikić, M. Fast and accurate de novo genome assembly from long uncorrected reads. *Genome Res.* **27**, 737–746 (2017).
9. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).
10. Li, H. & Durbin, R. Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* **25**, 1754–1760 (2009).
11. Broder, A. Z. On the resemblance and containment of documents. In *Proc. Compression and Complexity of SEQUENCES 1997* (*cat. no. 97TB100171*) (eds. Carpentieri, B. et al.) 21–29 (IEEE, 1997); https://doi.org/10.1109/SEQUEN.1997.666900
12. Jain, C., Dilthey, A., Koren, S., Aluru, S. & Phillippy, A. M. A fast approximate algorithm for mapping long reads to large reference databases. In *Research in Computational Molecular Biology* (ed. Sahinalp, S. C.) 66–81 (Springer, 2017).
13. Chin, C.-S. & Khalak, A. Human genome assembly in 100 minutes. Preprint at *bioRxiv* https://doi.org/10.1101/705616 (2019).
14. Fruchterman, T. M. J. & Reingold, E. M. Graph drawing by force-directed placement. *Softw. Pract. Exp.* **21**, 1129–1164 (1991).
15. Barnes, J. & Hut, P. A hierarchical *O*(*N*log*N*) force-calculation algorithm. *Nature* **324**, 446–449 (1986).
16. Wick, R. R. & Holt, K. E. Benchmarking of long-read assemblers for prokaryote whole genome sequencing. *F1000Res.* **8**, 2138 (2020).
17. Nurk, S. et al. HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. *Genome Res.* **30**, 1291–1305 (2020).
18. Cheng, H., Concepcion, G. T., Feng, X., Zhang, H. & Li, H. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nat. Methods* **18**, 170–175 (2021).
19. Belser, C. et al. Chromosome-scale assemblies of plant genomes using nanopore long reads and optical maps. *Nat. Plants* **4**, 879–887 (2018).
20. Choi, J. Y. et al. Nanopore sequencing-based genome assembly and evolutionary genomics of circum-basmati rice. *Genome Biol.* **21**, 21 (2020).
21. Vaser, R. & Šikić, M. Yet another de novo genome assembler. In *Proc. 2019 11th International Symposium on Image and Signal Processing and Analysis* (*ISPA*) (eds. Lončarić, S. et al.) 147–151 (IEEE, 2019); https://doi.org/10.1109/ISPA.2019.8868909
22. Simão, F. A., Waterhouse, R. M., Ioannidis, P., Kriventseva, E. V. & Zdobnov, E. M. BUSCO: assessing genome assembly and annotation completeness with single-copy orthologs. *Bioinformatics* **31**, 3210–3212 (2015).
23. Jain, M. et al. Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* **36**, 338–345 (2018).
24. Mikheenko, A., Prjibelski, A., Saveliev, V., Antipov, D. & Gurevich, A. Versatile genome assembly evaluation with QUAST-LG. *Bioinformatics* **34**, i142–i150 (2018).
25. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
26. Vaser, R. & Sikic, M. 2021. Assemblies generated in the manuscript ‘Time and memory efficient genome assembly with Raven’. Zenodo https://doi.org/10.5281/zenodo.4443062
27. Vaser, R. & Sikic, M. 2021. Raven source code used in the manuscript ‘Time and memory efficient genome assembly with Raven’. Zenodo https://doi.org/10.5281/zenodo.4672196

Acknowledgements

This work has been supported in part by the Croatian Science Foundation under the project ‘Single genome and metagenome assembly’ (IP-2018-01-5886, to M.Š.), the European Regional Development Fund under grant no. KK.01.1.1.01.0009 (DATACROSS, to M.Š.) and the A*STAR Computational Resource Centre through the use of their high-performance computing facilities. R.V. and M.Š. have been partially supported by funding from A*STAR, Singapore. We acknowledge Intel Corporation for allowing us to test with the Intel Optane persistent memory server and providing us with high-quality technical support. Finally, we thank G. Žužić from Carnegie Mellon University for valuable discussions about graph drawings.

Author contributions

M.Š. devised the project. R.V. designed and implemented Raven, and benchmarked it with other assemblers. Both authors drafted and revised the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-021-00073-4.

Correspondence and requests for materials should be addressed to M.Š.

Peer review information *Nature Computational Science* thanks the anonymous reviewers for their contribution to the peer review of this work. Handling editor: Ananya Rastogi, in collaboration with the *Nature Computational Science* team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.