NanoDJ: A Dockerized Jupyter Notebook for Interactive Oxford Nanopore MinION Sequence Manipulation and Genome Assembly

Héctor Rodríguez-Pérez1,§, Tamara Hernández-Beeftink1,§, José M. Lorenzo-Salazar2, José L. Roda-García3, Carlos J. Pérez-González4, Marcos Colebrook3*, Carlos Flores1,2,5*.

1Research Unit, Hospital Universitario Nuestra Señora de Candelaria, Universidad de La Laguna, Santa Cruz de Tenerife, Spain, 2Genomics Division, Instituto Tecnológico y de Energías Renovables (ITER), Santa Cruz de Tenerife, Spain, 3Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, Santa Cruz de Tenerife, Spain, 4Departamento de Matemáticas, Estadística e Investigación Operativa, Universidad de La Laguna, Santa Cruz de Tenerife, Spain, and 5CIBER de Enfermedades Respiratorias, Instituto de Salud Carlos III, Madrid, Spain.

*To whom correspondence should be addressed (mcolesan@ull.edu.es; cflores@ull.edu.es)

§These authors contributed equally.

Abstract

Background: The Oxford Nanopore Technologies (ONT) MinION portable sequencer makes it possible to use cutting-edge genomic technologies in the field and the academic classroom.

Results: We present NanoDJ, a Jupyter notebook integration of tools for simplified manipulation and assembly of DNA sequences produced by ONT devices. It integrates basecalling, read trimming and quality control, simulation and plotting routines with a variety of widely used aligners and assemblers, including procedures for hybrid assembly.

Conclusions: With the use of Jupyter-facilitated access to self-explanatory contents of applications and the interactive visualization of results, as well as by its distribution into a Docker software container, NanoDJ is aimed to simplify and make more reproducible ONT DNA sequence analysis. The NanoDJ package code, documentation and installation instructions are freely available at https://github.com/genomicsITER/NanoDJ.

Keywords: genome analysis, nanopore sequencing, Jupyter, Docker
**Background**

It has never been before so easy and affordable to access and utilize genetic variation of any organism and purpose. This has been motivated by the continuous development of high-throughput DNA sequencing technologies, most commonly known as Next Generation Sequencing (NGS). A key improvement is the possibility of obtaining long single molecule sequences with the fast and cost-efficiency technology released by Oxford Nanopore Technologies (ONT) and the marketing in 2014 of the MinION, a portable, pocket-size, nanopore-based NGS platform [1]. Since then, several algorithms and software tools have flourished specifically for ONT sequence data. Despite its size, it provides multi-kilobase reads with a throughput comparable to other benchtop sequencers in the market (1-10 Gbases by 2017), therefore still necessitating of efficient and integrated bioinformatics tools to facilitate the widespread use of the technology.

While MinION has shown promise in distinct applications [2], because of the low cost, laptop operability, and the USB-powered compact design of MinION, cutting-edge NGS technology is not any more necessarily linked to the established idea of a large machine with high cost that must be located in centralized sequencing centers or in a laboratory bench. As a consequence, the utility of MinION in field experiments to move from sample-to-answers on site have been demonstrated with infectious disease studies [3, 4], off-Earth genome sequencing [5], and species identification in extreme environments [6–8], among others. Leveraging of MinION capabilities in the academic classroom is a natural extension of these field studies to facilitate education of genomics in undergraduate and graduate students [9].

To date, there is no specific software solution aimed to facilitate ONT sequence analyses by integrating capabilities for data manipulation, sequence comparison and assembly in field experiments or for educational purposes to help facilitate learning of genomics [9]. We have developed NanoDJ, an interactive collection of Jupyter notebooks to integrate a variety of software, advanced computer code, and plain contextual explanations. In addition, NanoDJ is distributed as a Docker software container to simplify installation of dependencies and improve the reproducibility of results.
Features and functionalities

NanoDJ is distributed as a Docker container built underneath Jupyter notebooks, which is increasingly popular in life sciences to significantly facilitate the interactive exploration of data [10], and has been recently integrated in the widely used Galaxy portal [11]. The Docker container allows NanoDJ to run in an isolated, self-contained package, that can be executed seamlessly across a wide range of computing platforms [12], having a negligible impact on the execution performance [13]. NanoDJ integrates diverse applications (Supplementary Table S1) organized into 12 notebooks grouped on three sections (Fig. 1; Table 1). Main results are presented as embedded objects. In addition, one of the notebooks was conceived for educational purposes by setting a particularly simple problem and the inclusion of low-level explanations. To facilitate the use of the educational notebook and bypassing the installation of Docker and NanoDJ, a lightweight version of this notebook and small sets of ONT reads can be utilized from a web-browser using Binder (https://mybinder.org) in the NanoDJ GitHub repository. We illustrate the versatility of NanoDJ in distinct scenarios by providing results from four case studies (Supplementary Text 1).

Input, basecalling, and simulations

Input data can be a list of FAST5 files from previous basecalled runs (e.g. a Metrichor output) or event-level signal data to be basecalled using the latest ONT caller. The user can also simulate reads with NanoSim and pre-computed model parameters. This possibility is important in different scenarios as to help designing an experiment, or to bypass technical difficulties in academic setups [9].

Summary, quality control and filtering

Either for a simulated or an empirical run, the user will obtain summary data and plots informing of read length distribution, GC content vs. length, and read length vs. quality score (when available). If barcodes were used in the experiment, Porechop can be used for demultiplexing, barcode trimming and to filter out reads.

Genome assembly and comparison
Depending on the application, sequence data can be aligned against reference sequences or used for genome assembly using diverse methods. Alignment is performed either against one (BWA and Rebaler) or multiple (BLAST) reference sequences, providing the generation of BAM files for downstream applications (e.g., variant identification) or information of species composition. Alternatively, the user may opt for a de novo assembly. NanoDJ allows the use of some of the best-performing algorithms (Canu, Flye, and Miniasm), or to combine ONT reads with others obtained with second-generation NGS platforms for a hybrid assembly (Unicycler and MaSuRCA). The latter provides more effective assemblies and reduced error rate compared to assemblies based only on ONT reads [14]. NanoDJ includes the possibility of contig correction (Racon, Nanopolish, and Pilon). Assemblies can be evaluated with the embedded version of QUAST, and represented with Bandage.

**Limitations and future directions**

For non-expert users, it would have been better if NanoDJ was envisaged as an on-line application to facilitate its use. However, our main objective was to integrate major tools for the analysis of ONT sequences in an interactive software environment to facilitate learning the basics behind ONT sequence analysis while providing a useful tool for professionals. Providing it as a Dockerized solution simply bolsters the focus on the use of the tool, reducing the burden of installing all dependencies by the user. At the moment, NanoDJ is set for the analysis of small genomes and targeted NGS studies, although focusing on primary and secondary analysis of DNA sequences. The integration of tools for variant identification and tertiary analysis (annotation of variants or sequence elements, interpretation, etc.) [15-16], as well as for epigenetics [17] and direct RNA sequencing [18] will be the focus of further developments of NanoDJ.

**Conclusions**

We present NanoDJ as an integrated Jupyter-based toolbox distributed as a Docker software container to facilitate ONT sequence analysis. NanoDJ is best suited for the analyses of small genomes and targeted NGS
studies. We anticipate that the Jupyter notebook-based structure will simplify further developments in other applications.

**Availability and requirements**

**Project name**: NanoDJ

**Project home page**: https://github.com/

**Operating system(s)**: Windows, Linux, Mac OS

**Programming language**: Bash/Python

**Other requirements**: Docker installation

**License**: GPL

**Any restrictions to use by non-academics**: None

**Additional files**

- **Table S1**: Applications integrated in NanoDJ; **Supplementary Text 1**: Testing on case study datasets (**Table S2**: Datasets for illustrative uses of NanoDJ; **Table S3**: Comparison of *de novo* assemblies using different inputs or with an assembly corrector; **Table S4**: Comparison of three *de novo* assemblers in a high-coverage ONT dataset; **Table S5**: Comparison of results from two hybrid *de novo* assemblers; **Figure S1**: Human mitochondrial DNA variant representation against the reference sequence; **Table S6**: Source of mitochondrial DNA genomes, simulations and classification results.)

**Declarations**

**Ethics approval and consent to participate**

Not applicable

**Consent for publication**

Not applicable

**Availability of data and materials**

All data generated or analysed during this study are included in this published article and its supplementary information files.

**Funding**
This research was funded by the Instituto de Salud Carlos III (grants PI14/00844 and PI17/00610), the Spanish Ministry of Science, Innovation and Universities (grant RTC-2017-6471-1; MINECO/AEI/FEDER, UE), the Spanish Ministry of Economy and Competitiveness (grant MTM2016-74877-P), which were co-financed by the European Regional Development Funds ‘A way of making Europe’ from the European Union, and by the agreement OA17/008 with Instituto Tecnológico y de Energías Renovables (ITER) to strengthen scientific and technological education, training, research, development and innovation in Genomics, Personalized Medicine and Biotechnology.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

HRP scripted and tested the software, and contributed to data analysis; THB was involved in data analysis and interpretation; JLS was involved in data analysis; JRG and CPG revised and tested the software, and revised the manuscript; MC conceived the project, revised and tested the software, and revised the manuscript; CF conceived the project, designed the software, interpreted the data, and critically revised the manuscript.

Acknowledgements

Not applicable
References

1. Brown CG, Clarke J: Nanopore development at Oxford Nanopore. Nat. Biotechnol. 2016, 34:810–811.

2. Jain M, Olsen HE, Paten B, Akeson M: The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. Genome Biol. 2016, 17:239.

3. Quick J, Loman NJ, Duraffour S, et al.: Real-time, portable genome sequencing for Ebola surveillance. Nature 2016, 530:228–232.

4. Faria NR, Quick J, Claro IM, Thézé J, de Jesus JG, Giovanetti M, Kraemer MUG, Hill SC, Black A, da Costa AC, Franco LC, Silva SP, Wu C-H, Raghwani J, Cauchemez S, du Plessis L, Verotti MP, de Oliveira WK, Carmona EH, Coelho GE, Santelli ACFS, Vinhal LC, Henrriques CM, Simpson JT, Loose M, Andersen KG, Grubaugh ND, Somasekar S, Chiu CY, Muñoz-Medina JE, Gonzalez-Bonilla CR, Arias CF, Lewis-Ximenez LL, Baylis SA, Chieppe AO, Aguiar SF, Fernandes CA, Lemos PS, Nascimento BLS, Monteiro HAO, Siqueira IC, de Queiroz MG, de Souza TR, Bezerra JF, Lemos MR, Pereira GF, Loudal D, Moura LC, Dhalia R, França RF, Magalhães T, Marques ET Jr, Jaenisch T, Wallau GL, de Lima MC, Nascimento V, de Cerqueira EM, de Lima MM, Mascarenhas DL, Neto JPM, Levin AS, Tozetto-Mendoza TR, Fonseca SN, Mendes-Correa MC, Milagres FP, Segurado A, Holmes EC, Rambaut A, Bedford T, Nunes MRT, Sabino EC, Loman NJ, Pybus OG: Establishment and cryptic transmission of Zika virus in Brazil and the Americas. Nature 2017, 546:406–410.

5. Castro-Wallace SL, Chiu CY, John KK, Stahl SE, Rubins KH, McIntyre ABR, Dworkin JP, Lupisella ML, Smith DJ, Botkin DJ, Stephenson TA, 5. Stephenson TA, Juul S, Turner DJ, Izquierdo F, Federman S, Stryke D, Somasekar S, Alexander N, Yu G, Mason C, Burton AS: Nanopore DNA Sequencing and Genome Assembly on the International Space Station. Scientific Reports 2017, 18022(7):1

6. Johnson SS, Zaikova E, Goerlitz DS, Bai Y, Tighe SW: Real-Time DNA Sequencing in the Antarctic Dry Valleys Using the Oxford Nanopore Sequencer. J Biomol Tech. 2017, 28(1):2-7.

7. Pomerantz A, Peñafiel N, Arteaga A, Bustamante L, Pichardo F, Coloma LA, Barrio-Amoros CL, Salazar-Valenzuela D, Prost S: Real-time DNA barcoding in a remote rainforest using nanopore sequencing. Gigascience 2018, 7(4):giy033.

8. Menegon M, Cantaloni C, Rodriguez-Prieto A, Centomo C, Abdelfattah A, Rossato M, Bernardi M, Xumerle L, Loader S, Delledonne M: On site DNA barcoding by nanopore sequencing. PLoS One 2017, 12:e0184741.

9. Zaaijer S, Columbia University Ubiquitous Genomics 2015 class, Erlich Y: Using mobile sequencers in an academic classroom. Elife. 2016, 5:e14258.

10. Almugbel R, Hung LH, Hu J, Almutairi A, Ortogero N, Tamta Y, Yeung KY: Reproducible Bioconductor workflows using browser-based interactive notebooks and containers. J. Am. Med. Inform. Assoc. 2018, 25, 4–12.

11. Grüning BA, Rasche E, Rebolledo-Jaramillo B, Eberhard C, Houwaart T, Chilton J, Coraor N, Backofen R, Taylor J, Nekrutenko A: Jupyter and Galaxy: Easing entry barriers into complex data analyses for biomedical researchers. PLoS Comput. Biol. 2017, 13, e1005425.

12. Boettiger C: An Introduction to Docker for Reproducible Research. Oper. Syst. Rev. 2015, 49, 71–79.

13. Di Tommaso P, Palumbo E, Chatzou M, Prieto P, Heuer ML, Notredame C: The impact of Docker containers on the performance of genomic pipelines. PeerJ 2015, 3, e1273.

14. Wick RR, Judd LM, Gorrie CL, Holt KE: Completing bacterial genome assemblies with multiplex MinION sequencing. Microb Genom. 2017, 3(10):e000132.
15. Cook D, Valle-Inclan JE, Pajoro A, Rovenich H, Thomma B, Faino L: Long Read Annotation (LoReAn): automated eukaryotic genome annotation based on long-read cDNA sequencing. *bioRxiv*. 2017.

16. Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, Schatz MC: Accurate detection of complex structural variations using single-molecule sequencing. *Nature Methods*. 2018, 15: 461-468.

17. Stoiber MH, Quick J, Egan R, Lee JE, Celniker SE, Neely R, Loman N, Pennacchio L, Brown JB: De novo Identification of DNA Modifications Enabled by Genome-Guided Nanopore Signal Processing. *bioRxiv*. 2016.

18. Garalde DR, Snell EA, Jachimowicz D, Sipos B, Lloyd JH, Bruce M, Pantic N, Admassu T, James P, Warland A, Jordan M, Ciccone J, Serra S, Keenan J, Martin S, McNeill L, Wallace EJ, Jayasinghe L, Wright C, Blasco J, Young S, Brocklebank D, Juul S, Clarke J, Heron AJ, Turner DJ: Highly parallel direct RNA sequencing on an array of nanopores. *Nature Methods*. 2018, 15:201-206.
Fig. 1. Simplified scheme of all NanoDJ functionalities.

Table 1. Summary of NanoDJ notebooks.

| Name                                | Functionality                                                                 |
|-------------------------------------|-----------------------------------------------------------------------------|
| 0.0_QualityControl.ipynb           | Evaluate the quality control and sequence handling                          |
| 1.0_Basecalling.ipynb              | Translates the events or the raw electrical signal from an ONT sequencer (FAST5 format) to a DNA sequence to obtain a FASTA or a FASTQ file |
| 1.1_Trim+Demux.ipynb              | Perform sequence trimming and demultiplexing                                |
| 2.0_DeNovo_Canu-Miniasm.ipynb      | De novo assembly with Canu or Miniasm, and polish with Racon and Pilon     |
| 3.0_DeNovo_Canu+polish.ipynb       | Nanopolish modules to improve the Canu assembly                             |
| 4.0_Denoovo_Flye.ipynb             | De novo assembly with Flye software                                        |
| 5.0_DeNovo_Hybrid.ipynb            | Perform de novo assembly of Nanopore reads in conjunction with Illumina reads using MaSuRCA and/or Unicycler software |
| 6.0_AssemblyCompare.ipynb          | Compare distinct assembly results based on QUAST software                   |
| 7.0_SimulateReads.ipynb            | Obtain simulated reads made with Nanosim software and the Nanosim-h fork with precomputed models |
| 8.0_Alignment.ipynb                | Reference-based assembly using either BWA, BLAST or Rebaler software        |
| 9.0_AssemblyGraph.ipynb            | Assembly graph visualization                                               |
| Educational.ipynb                  | Performs basecalling (with Albacore), quality control steps, and a BLAST-based classification of the reads (for educational purposes) |
**Table S1.** Applications integrated in NanoDJ.

| Application      | Process                                      | Result                                      |
|------------------|----------------------------------------------|---------------------------------------------|
| Albacorea v1.0.1 | Basecalling                                   | FASTQ file                                 |
| NanoSim-hb v1.0.0.3 | Read simulation                             | FASTA file with simulated reads             |

**SUMMARY, QUALITY CONTROL AND FILTERING**

| Application      | Process                                      | Result                                      |
|------------------|----------------------------------------------|---------------------------------------------|
| Biopython v1.70  | Summarization                                | Plots and tables                            |
| Porechop v0.2.2  | Demultiplexing, trimming and filtering       | Demultiplexed FASTQ files                   |

**GENOME ASSEMBLY AND COMPARISONS**

| Application      | Process                                      | Result                                      |
|------------------|----------------------------------------------|---------------------------------------------|
| BWA v0.7.17      | Alignment (one reference)                    | BAM file                                   |
| Rebale v0.1.0    | Alignment (one reference)                    | FASTA file                                 |
| BLAST v2.7.1     | Alignment (multiple reference)               | Read assignment (summary)                   |
| Miniasm v1.04    | De novo assembly                             | Assembly files (FASTA, PAF)                 |
| Flye v2.3.1      | De novo assembly                             | Assembly files (FASTA)                      |
| Canu v1.6        | De novo assembly                             | Assembly files (FASTA, .gfa)                |
| Racon v0.5.0     | Contig correction                            | Corrected FASTA                            |
| Nanopolish v0.8.5| Contig correction                            | Corrected FASTA                            |
| Pilon v1.22      | Contig correction                            | Corrected FASTA                            |
| Unicycler v0.4.1 | Hybrid de novo assembly                      | Assembly files (FASTA, .gfa)                |
| MaSuRCA v3.2.2   | Hybrid de novo assembly                      | Assembly files (FASTA, .gfa)                |
| QUAST v5.0.0     | Assembly comparison                          | Summary tables and plots                    |
| Bandage v0.8.1   | Visualization of assembly graphs (.gfa)      | Assembly plot                               |
Supplementary Text 1. Testing on case study datasets

We illustrate the versatility of NanoDJ in distinct scenarios with four example datasets starting from diverse file inputs (Supplementary Table S2): (1) the assembly of a bacterial genome testing distinct de novo assemblers based on a high-coverage ONT data (different inputs); (2) a hybrid genome assembly of a bacterial genome based on low-pass ONT sequencing and short-read data at high coverage; (3) an emulation of a resequencing experiment to map ONT reads to a reference sequence for the identification of genetic variants (with third-party tools); and (4) an evaluation of species composition based on simulated ONT reads. All analyses were performed in a Ubuntu 16.04 Server with two Intel Xenon E5-2650 12-core 2.2 GHz processors and 256 Gb of RAM.

Table S2. Datasets for illustrative uses of NanoDJ.

| Example | Organism      | Size (Mb) | Dataset | File(s) | Sequencing | #reads (x1000) | Source     |
|---------|---------------|-----------|---------|---------|------------|----------------|------------|
| 1       | *E. coli*     | 4.6       | R9      | Fast5/Fasta | Rapid 1D   | 164.47         | Web⁹       |
| 2       | *S. agalactiae* | 2.2       | R9/MiSeq | Fast5/Fastq | Rapid 1D/PE300 | 3.72/2.975 | This study⁸ |
| 3       | *H. sapiens* (mitochondria) | 0.016     | R9.4    | Fast5    | Rapid 1D   | 59.36          | Web⁵       |
| 4       | Vertebrates (mitochondria) | Variable | .       | Fasta    | Simulated  | 0.21           | Web⁴       |

Example 1. We used NanoDJ to test different de novo assemblers using default parameters in a 4.6 Mb reference *E. coli* genome (K-12-MG1655; NC_000913.3) obtained at a high coverage (>20X) with ONT. For these analyses we have used the following notebooks in the order: 1st) 1.0_Basecalling.ipynb, 2nd) 2.0_DeNovo_Canu-Miniasm.ipynb, 3rd) 4.0_DeNovo_Flye.ipynb, and 4th) 6.0_AssemblyCompare.ipynb. A first test with the most accurate assembler (Canu) compared the use of FAST5 files (333 Gb) and of a FASTA (1.5 Gb) file as the input dataset. In both cases, Canu assembled all reads into a single 4.6 Mb genome covering 99.9% of the reference. Using FASTA as the input did not reduce the elapsed time and introduced more mismatches and indels (Supplementary Table S3). The use of assembly polishers (e.g. Racon) did not improve significantly the assembly. Besides, we used the FASTA file to compare three distinct assemblers (Canu, Flye, and Miniasm). Overall, Canu and Flye were by far the most accurate alternatives, but Flye was much less computationally expensive. Miniasm was able to assemble the genome into a single contig, and was the fastest alternative of all three (Supplementary Table S4).

Table S3. Comparison of Canu-based de novo assemblies of *E. coli* K-12-MG1655⁵ using different inputs or with an assembly corrector.

| Parameters                          | FASTA | FAST5 | FAST5-Racon |
|-------------------------------------|-------|-------|-------------|
| Total length assembled (bp)         | 4,602,643 | 4,662,047 | 4,706,877   |
| Contigs (>500 bp)                    | 1     | 1     | 1           |
| Genome fraction (%)                  | 99.88 | 99.98 | 99.99       |

*Dependencies included: Numpy, Matplotlib and Pandas.
**Dependencies included: Spades, Racon, Pilon, bowtie2, samtools, blast+. 

---

¹https://nanoporetech.com/
²Yang et al., 2017 (https://pypi.python.org/pypi/NanoSim-H)
³Cock et al., 2009 (www.biopython.org)
⁴https://github.com/rrwick/Porechop
⁵Li & Durbin 2010 (https://pypi.python.org/pypi/NanoSim-H)
⁶https://github.com/rrwick/Rebaler
⁷Altschul et al., 1990 (ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST)
⁸Lin et al., 2016 (https://github.com/rrwick/Unicycler)
⁹Koren et al., 2017 (https://github.com/nanopore-wgs-consortium/NA12878)
¹⁰Vaser et al., 2017 (https://github.com/isovic/racon)
¹¹https://github.com/nanopolish
¹²Walker et al., 2014 (https://github.com/broadinstitute/panic)
¹³Wick et al., 2017 (https://github.com/nanopore-wgs-consortium/NA12878)
¹⁴Cock et al., 2009 (www.biopython.org)
¹⁵https://github.com/rrwick/Unicycler
¹⁶https://github.com/rrwick/Rebaler
¹⁷Li & Durbin 2010 (https://pypi.python.org/pypi/NanoSim-H)
¹十八NCBI Sequence Read Archive IDs: SRP141332 (ONT), SRP141319 (Illumina)
¹⁹http://lab.loman.net/2016/07/30/nanopore-r9-data-release
²⁰http://hgdownload.soe.ucsc.edu/downloads.html
²¹http://hgdownload.soe.ucsc.edu/downloads.html
²²Mb: Megabases
| Parameters                | Canu | Flye | Miniasm |
|---------------------------|------|------|---------|
| Total length assembled (bp) | 4,602,643 | 4,678,264 | 4,404,394 |
| Contigs (≥500 bp)         | 1    | 1    | 1       |
| Largest alignment (pb)*    | 3,336,128 | 2,310,685 | 77      |
| Mismatches (#)*           | 14,674 | 12,678 | 0       |
| Indels (#)*               | 55,889 | 40,090 | 2       |
| N50                       | 4,602,643 | 4,678,264 | 4,404,394 |
| GC (%)                    | 51.07 | 50.72 | 52.48   |
| Elapsed time (sec.)       | 138,221.92 | 1,038  | 104     |

Table S4. Comparison of three de novo assemblers in a high-coverage ONT dataset (FASTA input) obtained from E. coli K-12-MG1655.  

Table S5. Comparison of results from two hybrid de novo assemblers in a S. agalactiae dataset.

### Table S4. Comparison of three de novo assemblers in a high-coverage ONT dataset (FASTA input) obtained from E. coli K-12-MG1655.

**Parameters**

| Parameters                | Canu | Flye | Miniasm |
|---------------------------|------|------|---------|
| Total length assembled (bp) | 4,602,643 | 4,678,264 | 4,404,394 |
| Contigs (≥500 bp)         | 1    | 1    | 1       |
| Largest alignment (pb)*    | 3,336,128 | 2,310,685 | 77      |
| Mismatches (#)*           | 14,674 | 12,678 | 0       |
| Indels (#)*               | 55,889 | 40,090 | 2       |
| N50                       | 4,602,643 | 4,678,264 | 4,404,394 |
| GC (%)                    | 51.07 | 50.72 | 52.48   |
| Elapsed time (sec.)       | 138,221.92 | 1,038  | 104     |

Genome size and GC content are 4,641,652 bp and 50.79%, respectively.  
*Against E. coli reference sequence NC_000913.3.  
Indels: insertion/deletion variants; N50: minimum contig length to cover at least 50% of the genome; GC: guanine-cytosine content.

### Table S5. Comparison of results from two hybrid de novo assemblers in a S. agalactiae dataset.

**Parameters**

| Parameters                | Unicycler | MaSuRCA |
|---------------------------|-----------|----------|
| Total length assembled (bp) | 2,159,288 | 2,099,176 |
| Contigs (≥500 bp)         | 8         | 16       |
| Largest contig (bp)       | 1,019,216 | 627,716  |
| Genome fraction (%)*      | 98.57     | 95.29    |

Genome size and GC content are 4,641,652 bp and 50.79%, respectively.  
*Against E. coli reference sequence NC_000913.3.  
Indels: insertion/deletion variants; N50: minimum contig length to cover at least 50% of the genome; GC: guanine-cytosine content.

**Example 2.** Here we used available S. agalactiae reads from a low-pass (~2.5X) ONT MinION experiment and 2.97 million reads (>300X) from a MiSeq (Illumina, Inc.) run to compare two state-of-the-art hybrid assemblers. In this case, we have used the following notebooks in the order: 1) 1.0_Basecalling.ipynb, 2) 5.0_DeNovo_Hybrid.ipynb, 3) 6.0_AssemblyCompare.ipynb, and 4) 9.0_AssemblyGraph.ipynb. NanoDJ integrated basecalling with Albacore, the hybrid assembly with Unicycler or MaSuRCA, and the assembly comparisons. Unicycler was superior to MaSuRCA based on the number of contigs, N50, the size of the largest contig, the number of mismatches and indels, and the proportion of covered genome (Supplementary Table S5). Plotting of Unicycler results also allowed isolating a 2.5 Kb plasmid sequence supported by our previous findings (unpublished), while MaSuRCA split its sequence into two small contigs.
**Example 3.** We used FAST5 files available for reads mapping to the human mitochondrial DNA (33 Gb) from the NA12878 human genome reference standard on the ONT MinION. NanoDJ was used to extract all reads and obtain a FASTQ file, integrate all reads into a single FASTA assembly by mapping against the revised Cambridge Reference Sequence (NC_012920, gi:251831106) with Rebaler, and obtain a BAM file, which was then manually inspected with IGV (Robinson et al., 2011) (Supplementary Fig. S1). In this case study, we have used the following notebooks in the order: 1st) 1.0_Basecalling.ipynb, 2nd) 0.0_QualityControl.ipynb, and 3rd) 8.0_Alignment.ipynb. The assembly was then used in HAPLOFIND, a third-party tool based on PhyloTree build 17 (Vianello et al., 2013), to confirm the classification of the reference data as H13a1a1 mitochondrial haplogroup.

**Example 4.** We used NanoDJ to create a local BLAST database with seven mitochondrial reference genomes from distinct vertebrates. In parallel, NanoDJ was used to simulate 30 ONT reads from seven species with NanoSim-h, which were merged into a single FASTA to emulate a 210-read heterogeneous sample run with balanced proportion of species (~14.3% each). NanoDJ was finally used for a BLAST-based classification of simulated reads to obtain species abundance. In this case study, we have used the following notebooks in the order: 1st) 7.0_SimulateReads.ipynb, 2nd) 0.0_QualityControl.ipynb, and 3rd) 8.0_Alignment.ipynb. As expected, the average abundance was supported by 14.3% of reads (excluding the unassigned), although with minimal fluctuations between 12.9 and 16.1% (Supplementary Table S6).

---

### Table

| Description                  | Value 1 | Value 2 |
|------------------------------|---------|---------|
| Largest alignment (pb)*      | 1,018,789 | 627,116  |
| Mismatches (#)*              | 165     | 230     |
| Indels (#)*                  | 38      | 53      |
| N50                          | 714,293 | 434,380 |
| GC (%)                       | 35.39   | 35.47   |
| Elapsed time (sec.)          | 9,836.52 | 3,009.03 |

---

*Streptococcus agalactiae* sequencing data was generated in house as part of an independent study using the Rapid Sequencing kit (SQK-RAD001) (Oxford Nanopore Technologies Ltd., Oxford, UK) in a single MinION 22-h run. Paired-end 300 bp reads for the same isolate were obtained using Nextera XT kit in a MiSeq Reagent kit V3 (Illumina, Inc., San Diego, CA), following the manufacturer’s recommendations.

*Mismatches: insertion/deletion variants; N50: minimum contig length to cover at least 50% of the genome; GC: guanine-cytosine content

---

**Figure S1.** Human mitochondrial DNA variant representation against the reference sequence (left), read distribution (right), and coverage information (bottom). A consensus calling of variant alleles was made if the allele was present in >70% of reads. This identified the correct nucleotide at all reference positions, permitting the identification of the expected haplogroup for the reference DNA. The figure illustrates the case of position 4,745 where a A>G change is supported by the reads, defining H13a1a1 haplogroup.
Table S6. Source of mitochondrial DNA genomes, simulations and classification results.

| Species                  | Reference                                      | Simulated reads | Assigned reads | Proportion* |
|--------------------------|-----------------------------------------------|-----------------|----------------|-------------|
| Gallus gallus            | Dec. 2015 (Gallus_gallus-5.0/galGal5)          | 30              | 27             | 14.5        |
| Alligator mississippiens | Aug. 2012 (allMis0.2/allMis1)                 | 30              | 25             | 13.4        |
| Bos taurus               | Jun. 2014 (UMD_3.1.1/bosTau8)                 | 30              | 27             | 14.5        |
| Equus caballus           | Sep. 2007 (Broad/equCab2)                     | 30              | 29             | 15.6        |
| Oreochromis niloticus    | Jan. 2011 (Nile tilapia/oreNil2)              | 30              | 30             | 16.1        |
| Rattus norvegicus        | Jul. 2014 (RGSC 6.0/rn6)                      | 30              | 24             | 12.9        |
| Ovis aries               | Aug. 2012 (ISGC Oar_v3.1/oviAri3)             | 30              | 24             | 12.9        |

*Excluding 24 reads that were unassigned.

Supplementary References

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: Basic local alignment search tool. J. Mol. Biol. 1990, 215, 403–410.

Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, de Hoon MJ: Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009, 25, 1422–1423.

Gurevich A, Saveliev V, Vyahhi N, Tesler G: QUAST: quality assessment tool for genome assemblies. Bioinformatics. 2013, 29, 1072–1075.

Li: Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. Bioinformatics. 2016, 32, 2103–2110.

Li & Durbin: Fast and accurate long-read alignment with Burrows-Wheeler Transform. Bioinformatics. 2010, 26, 589–595.

Kolmogorov M, Yuan J, Lin Y, Pevzner P: Assembly of long error-prone reads using repeat graphs. 2018, https://doi.org/10.1101/247148.

Sergey Koren, Brian P. Walenz, Konstantin Berlin, Jason R. Miller, Adam M. Phillippy: Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. Genome Res. 2016, 27, 722–736.

Robinson JT, Thorvaldsdóttir H, Winckler W, Guttman M, Lande ES, Getz G, Mesirov JP: Integrative genomics viewer. Nat. Biotechnol. 2011, 29, 24–26.

Vaser R, Sović I, Nagarajan N, Šikić M: Fast and accurate de novo genome assembly from long uncorrected reads. Genome Res. 2017, 27, 737–746.

Vianello D, Sevini F, Castellani G, Lombardi L, Capri M, Franceschi C: HAPLOFIND: a new method for high-throughput mtDNA haplogroup assignment. Hum. Mutat. 2013, 34, 1189–1194.

Walker BJ, Abeel T, Shea T, Priest M, Abouelliel A, Sankhyan S, Cuomo CA, Zeng Q, Wortman J, Young SK, Earl AM: Pilon: An integrated tool for comprehensive microbial variant detection and genome assembly improvement. PLoS ONE. 2014, 9, e112963.

Wick RR, Judd LM, Gorrie CL, Holt KE: Unicycler: Resolving bacterial genome assemblies from short and long sequencing reads. PLoS Comput. Biol. 2017, 13, e1005595.

Wick RR, Schultz MB, Zobel J, Holt KE: Bandage: interactive visualization of de novo genome assemblies. Bioinformatics. 2015, 31, 3350–3352.

Yang C, Chu J, Warren RL, Birol I: NanoSim: nanopore sequence read simulator based on statistical characterization. Gigascience. 2017, 6, 1-6.

Zimin AV, Marçais G, Puiu D, Roberts M, Salzberg SL, Yorke JA: The MaSuRCA genome assembler. Bioinformatics. 2013, 29, 2669–2677.