Bioinformatics, 34(15), 2018, 2666–2669
doi: 10.1093/bioinformatics/bty149
Advance Access Publication Date: 14 March 2018
Applications Note

Sequence analysis

NanoPack: visualizing and processing long-read sequencing data

Wouter De Coster[1,*], Svenn D'Hert[2], Darrin T. Schultz[3], Marc Cruts[1] and Christine Van Broeckhoven[1]

[1]Neurodegenerative Brain Diseases Group, [2]Bioinformatics, Neuromics Support Facility, Center for Molecular Neurology, VIB & University of Antwerp, 2610 Antwerp, Belgium and [3]Department of Biomolecular Engineering and Bioinformatics, University of California Santa Cruz, Santa Cruz, CA 95064, USA

*To whom correspondence should be addressed.

Associate Editor: Bonnier Berger

Received on January 2, 2018; revised on February 19, 2018; editorial decision on March 6, 2018; accepted on March 12, 2018

Abstract

Summary: Here we describe NanoPack, a set of tools developed for visualization and processing of long-read sequencing data from Oxford Nanopore Technologies and Pacific Biosciences.

Availability and implementation: The NanoPack tools are written in Python3 and released under the GNU GPL3.0 License. The source code can be found at https://github.com/wdecoster/nanopack, together with links to separate scripts and their documentation. The scripts are compatible with Linux, Mac OS and the MS Windows 10 subsystem for Linux and are available as a graphical user interface, a web service at http://nanoplot.bioinf.be and command line tools.

Contact: wouter.decoster@molgen.vib-ua.be

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The dominant sequencing by synthesis technology is characterized by sequencing a fixed short read length template (50–300 bp) with high accuracy (error rate <1%) (Goodwin *et al.*, 2016). In contrast, long-read sequencing methods from Oxford Nanopore Technologies (ONT) and Pacific Biosciences routinely achieve read lengths of 10 kb, with a long tail of up to 1.2 Megabases for ONT (unpublished results). These long reads come with a tradeoff of lower accuracy of about 85–95% (Giordano *et al.*, 2017; Jain *et al.*, 2017, 2018). It is evident that these characteristics make many existing Illumina-tailored QC tools, such as FastQC (Babraham Bioinformatics 2010, https://www.bioinformatics.babraham.ac.uk/projects/fastqc/), suboptimal for long-read technologies. NanoPack, a set of Python scripts for visualizing and processing long-read sequencing data, was developed to partially bridge this gap. Earlier tools such as poretools (Loman and Quinlan, 2014), poRe (Watson *et al.*, 2015) and IONiseR (Smith, 2017) mainly focused on feature extraction from the older fast5 file formats, and alternative tools such as pycoQC (Leger, 2017) and minion_qc (Lanfear, n.d. https://github.com/roblanf/minion_qc) do not offer the same flexibility and options as NanoPack. The plotting style from the pauvre tool (Schultz, n.d. https://github.com/conchoecia/pauvre) got incorporated in NanoPack (Supplementary Fig. S3).

2 Software description

2.1 Installation and dependencies

NanoPack and individual scripts are available through the public software repositories PyPI using pip and bioconda through conda (Dale *et al.*, 2017). The scripts build on a number of third party Python modules: matplotlib (Hunter, 2007), pysam (Heger, 2009; Li *et al.*, 2009; https://github.com/pysam-developers/pysam), pandas (McKinney, 2011), numpy (Walt *et al.*, 2011), seaborn (Waskom *et al.*, 2017) and biopython (Cock *et al.*, 2009).

2.2 Scripts for statistic evaluation and visualization

NanoStat produces a comprehensive statistical data summary (Supplementary Table S2). NanoPlot and NanoComp produce informative QC graphs displaying multiple aspects of sequencing data (Fig. 1, Supplementary Table S1) and accept input data in (compressed) fastq or fasta format, bam and (compressed) albacore summary files or multiple files of the same type.

All plots and summary statistics are combined in an html report. Because long and variable read lengths may be challenging to interpret on a linear axis, there is also an option to plot the read lengths on a log scale. Plots can be produced in standard image file formats including png, jpg, pdf and svg. NanoPlot produces read length histograms, cumulative yield plots, violin plots of read length

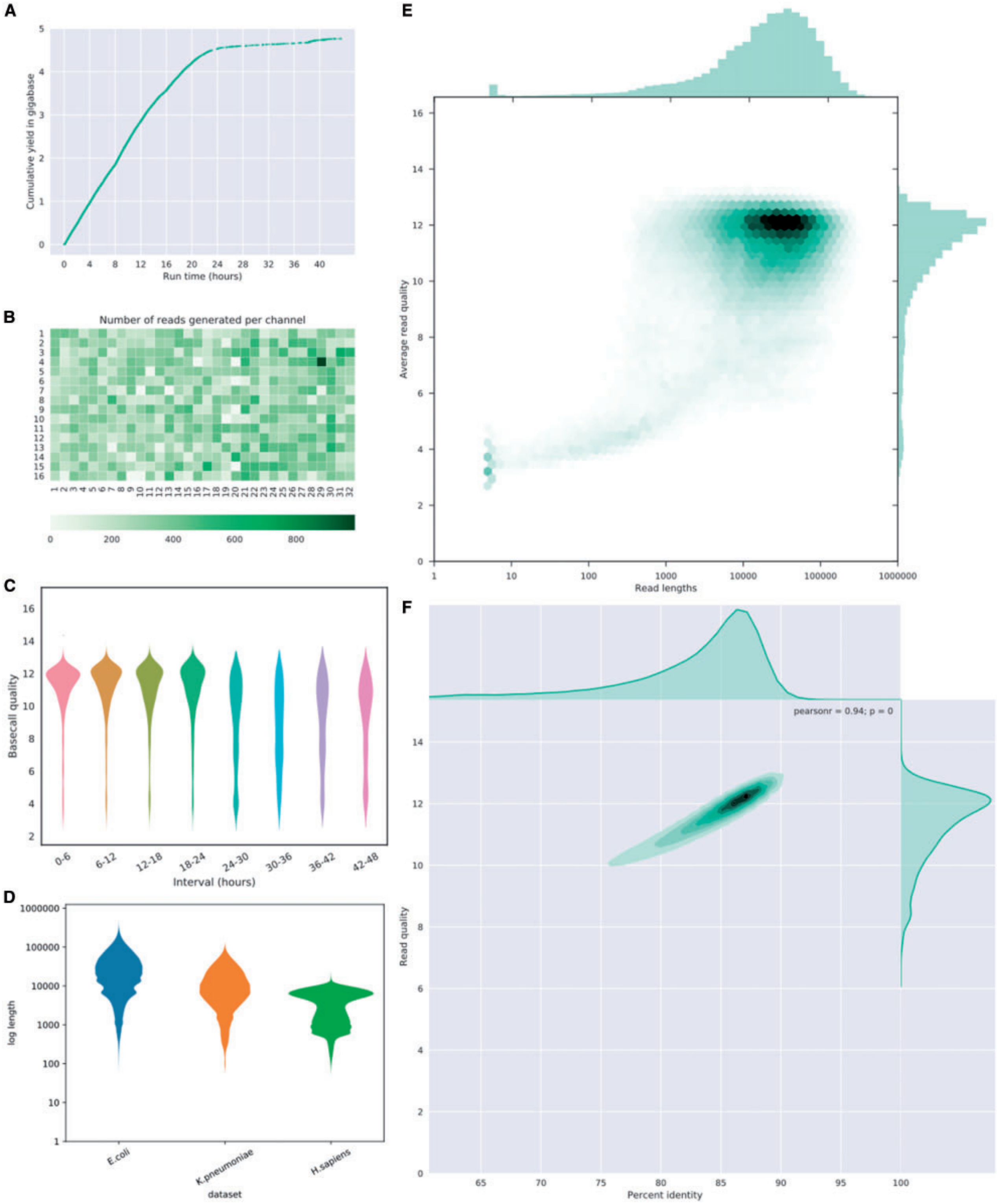

Fig. 1. Examples of plots of NanoPlot and NanoComp. (**A**) Cumulative yield plot (**B**) Flow cell activity heatmap showing number of reads per channel. (**C**) Violin plots comparing base call quality over time. (**D**) NanoComp plot comparing log transformed read lengths of the *E.coli* dataset with a *K.pneumoniae* and human dataset. (**E**) Bivariate plot of log transformed read length against base call quality with hexagonal bins and marginal histograms. (**F**) Bivariate plot of base call quality against percent identity with a kernel density estimate and marginal density plots

and quality over time and bivariate plots comparing the relationship between read lengths, quality scores, reference identity and read mapping quality. Better insight in big datasets can be obtained using bivariate plots with a 2D kernel density estimation or hexagonal bins (Fig. 1E and F, Supplementary Fig. S3). Optional arguments include random down sampling of reads and removing all reads above a length cutoff or below a quality cutoff. Data from a multiplexed experiment in albacore summary format can be separated, resulting in plots and statistics per barcode. NanoComp performs comparison across barcodes or experiments of read length and quality distributions, number of reads, throughput and reference identity.

2.3 Scripts for data processing

NanoFilt and NanoLyse were developed for processing reads in streaming applications and therefore have a minimal memory footprint and can be integrated in existing pipelines prior to alignment. NanoFilt is a tool for read filtering and trimming. Filtering can be performed based on mean read quality, read length and mean GC content. Trimming can be done with a user-specified number of nucleotides from either read ends. NanoLyse is a tool for rapid removal of contaminant DNA, using the Minimap2 aligner through the mappy Python binding (Li, 2017). A typical application would be the removal of the lambda phage control DNA fragment supplied by ONT, for which the reference sequence is included in the package. However, this approach may lead to unwanted loss of reads from regions highly homologous to the lambda phage genome.

3 Examples and discussion

The NanoPlot and NanoComp examples (Fig. 1) are based on an ONT *Escherichia coli* dataset from an ultra-long-read protocol sequenced on an R9.4 MinION flow cell (Quick and Loman, 2017; http://lab.loman.net/2017/03/09/ultrareads-for-nanopore/) generating 150 735 reads, base called using Albacore 2.0.2 and aligned to the *E.coli* reference genome using Minimap2 (Li, 2017). The cumulative yield (Fig. 1A) shows a lower efficiency when the flow cell wears out. A heat map of the physical layout of the MinION flow cell (Fig. 1B) highlights more productive channels and could potentially identifying suboptimal loading conditions, such as introduction of an air bubble. The mean base call quality per 6 h interval (Fig. 1C) shows a uniform high quality in the beginning, with lower quality reads after 24 h. In a bivariate plot comparing log transformed read lengths with their mean quality score (Fig. 1E) the majority of reads can be identified at lengths of 10 kb and quality scores of 12 by the color intensity of the hexagonal bins, with a subgroup of low-quality short reads. Plotting the mean quality against the per read percent reference identity (as a proxy for accuracy) (Fig. 1F) highlights a strong correlation, here with the number of reads plotted using a kernel density estimate. Additional examples from NanoPlot can be found in the supplementary information online, including standard and log transformed histograms, optionally with the N50 metric (Supplementary Figs S1 and S2) and a bivariate plot comparing effective read length with aligned read length (Supplementary Fig. S4), identifying reads which are only partially aligned to the reference genome.

The NanoComp plot (Fig. 1D) compares the log transformed read lengths of the same *E.coli* dataset to a *Klebsiella pneumoniae* (Wick *et al.*, 2017) and a human PromethION dataset (unpublished), clearly showing differences in the length profile with far longer reads in the *E.coli* dataset, standard read lengths in the library prep by ligation from *K.pneumoniae* and suboptimal read lengths from the human sample. Additional examples from NanoComp can be found in the supplementary information online, indicating that the *K.pneumoniae* library has both the highest yield (Supplementary Fig. S5) and on average higher quality scores (Supplementary Fig. S6) than both the human and *E.coli* dataset, but a comparable percent identity (Supplementary Fig. S7) with the human dataset.

4 Conclusion

NanoPack is a package of efficient Python scripts for visualization and processing of long-read sequencing data available on all major operating systems. Installation from the PyPI and bioconda public repositories is trivial, automatically taking care of dependencies. The plotting tools are flexible and customizable to the users need. Using a single NanoPlot or NanoComp command a full html report containing all summary statistics and plots can be prepared, and the software is easily accessible through the graphical user interface and web service, in addition to the command line scripts.

Acknowledgements

The authors acknowledge Mick Watson for contributing the mean GC content filtering to NanoFilt and Andreas Sjödin for maintaining the bioconda build recipes. The authors are also thankful to the many users who provided helpful suggestions and feature requests for these scripts. They thank the anonymous reviewers for helpful comments to improve our tools and this article.

Funding

The study was in part funded by the VIB (Flanders Institute for Biotechnology, Belgium), the University of Antwerp and the Flanders Agency for Innovation and Entrepreneurship (VLAIO). W.D.C. is a recipient of a PhD fellowship from VLAIO. D.T.S. is supported by NSF DGE [1339067].

Conflict of Interest: none declared.

References

Babraham Bioinformatics and Andrews,S. 2010. FastQC A quality control tool for high throughput sequence data. April 26, 2010. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

Cock,P.J.A. *et al.* (2009) Biopython: freely Available Python Tools for Computational Molecular Biology and Bioinformatics. *Bioinformatics*, **25**, 1422–1423.

Dale,R. *et al.*; The Bioconda Team, and Johannes Köster. (2017). Bioconda: a sustainable and comprehensive software distribution for the life sciences. bioRxiv. doi.org/10.1101/207092.

Giordano,F. *et al.* (2017) De novo yeast genome assemblies from MinION, PacBio and MiSeq platforms. *Sci. Rep.*, **7**, 3935.

Goodwin,S. *et al.* (2016) Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.*, **17**, 333–351.

Heger,A. 2009. Pysam. Github. https://github.com/pysam-developers/pysam.

Hunter,J.D. (2007) Matplotlib: a 2D graphics environment. *Comput. Sci. Eng.*, **9**, 90–95.

Jain,M. *et al.* 2018. Nanopore sequencing and assembly of a human genome with ultra-long reads. Nat. Biotechnol., January. doi.org/10.1038/nbt.4060.

Jain,M. *et al.* (2017) MinION analysis and reference consortium: phase 2 data release and analysis of R9.0 chemistry. *F1000Research*, **6**, 760.

Li,H. 2017. Minimap2: fast pairwise alignment for long DNA sequences. arXiv. https://arxiv.org/abs/1708.01492.

Li,H. *et al.*; 1000 Genome Project Data Processing Subgroup. (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

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

Leger,A. 2017. A-slide/pycoQC: v1.1.alpha2. doi.org/10.5281/zenodo.1116400. https://zenodo.org/record/1116400#.Wqo8pXvqG70.

McKinney,W. 2011. Pandas: a foundational python library for data analysis and statistics. Python for High Performance and Scientific Computing, 1–9. https://www.scribd.com/document/71048089/pandas-a-Foundational-Python-Library-for-Data-Analysis-and-Statistics.

Quick,J. and Loman,N. 2017. Thar She Blows! ultra long read method for nanopore sequencing · loman labs. 9 March 2017. http://lab.loman.net/2017/03/09/ultrareads-for-nanopore/.

Lanfear,R. n.d. Minion_qc. Github. https://github.com/roblanf/minion_qc. Accessed 13 February 2018.

Schultz,D. n.d. Pauvre. Github. https://github.com/conchoecia/pauvre. Accessed 13 February 2018.

Smith,M. 2017. IONiseR: quality assessment tools for Oxford Nanopore MinION Data. R Package ersion 2.2.0. http://bioconductor.org/packages/release/bioc/html/IONiseR.html.

Walt,S.V.D. *et al*. (2011) The NumPy array: a structure for efficient numerical computation. *Comput. Sci. Eng.*, **13**, 22–30.

Waskom,M. *et al*. 2017. Seaborn. July. zenodo.org/record/824567. https://zenodo.org/record/883859#Wqo20XaEfRY.

Watson,M. *et al*. (2015) poRe: an R package for the visualization and analysis of nanopore sequencing data. *Bioinformatics*, **31**, 114–115.

Wick,R.R. *et al*. (2017) Completing bacterial genome assemblies with multiplex MinION sequencing. *Microb. Genomics*, **3**, doi.org/10.1099/mgen.0.000132.