Proceedings of the 11th World Congress on Electrical Engineering and Computer Systems and Sciences (EECSS'25)
Paris, France – August, 2025
Paper No. ICBES 151
DOI: 10.11159/icbes25.151

NANOSLAST: Nanopore Signal Local Alignment and Segmentation Tool

Marketa Jakubickova[1], Michaela Zbudilova[1], Martin Suriak[1], Helena Vitkova[1]
[1]Department of Biomedical Engineering, Brno University of Technology
Technicka 12, Brno 61200, Czech Republic
jakubickova@vut.cz; xzbudi00@vutbr.cz; 240566@vut.cz; vitkovah@vut.cz

Abstract – Nanopore sequencing provided by Oxford Nanopore Technologies enables real-time analysis of native DNA or RNA molecules based on measured electrical signals called squiggles. While basecalling algorithms translate squiggles into nucleotide sequences, much of the raw signal information remains unexplored. Although several tools exist for inspecting the squiggles and offer simple signal visualization, they do not provide integrated workflows that allow users to search for specific nucleotide sequences and directly extract or visualize the corresponding raw signal segments. To address this gap, we developed NANOSLAST – a Nanopore Signal Local Alignment and Search Tool. This Python-based tool links local sequence similarity searches (via BLAST) to raw nanopore signals using basecalled SAM files and signal data from FAST5 or POD5 formats. The tool can be used to extract signals corresponding to specific motifs or functional elements, compare raw signal characteristics between different flowcells or sequencers or extract signal–sequence pairs suitable for training machine learning models focused on basecalling, methylation detection or other purposes.

***Keywords*:** nanopore sequencing; squiggle; signal processing; BLAST; pod5; fast5

1. Introduction

Nanopore sequencing, developed by Oxford Nanopore Technologies (ONT), has revolutionized the field of DNA/RNA sequencing with its unique capabilities. This technology enables rapid library preparation and real-time data analysis, and mainly it produces long-read, often tens to hundreds of kilobases in length [1], which enables accurate genome assembly. The principle of nanopore sequencing is based on measuring changes in ionic current, which are generated as nucleic acid strands passing through the pore located in the membrane to which voltage is applied [2]. The resulting signals, called squiggles, are then translated to nucleotide sequences using a process known as basecalling. However, decoding the bases is challenging, as the speed of strand passing through the pore is not constant, the measured current changes do not correspond to a single nucleotide, and nucleotides can be epigenetically modified, which also affects the signal [3], [4]. Therefore, complex neural networks are required for basecalling, making the process computationally and time demanding and despite the continuous development of new basecalling models, the error rate still ranges from 1-5% [5], [6]. Moreover, the basecalling process leads to loss of information as the raw signals contain more complexity, which cannot be preserved in the final nucleotide sequences. As a result, some biologically relevant information can be lost or misinterpreted during basecalling. Therefore, direct analysis and processing of the raw signal is necessary for some applications such as bacterial strain distinguishing [7], modified bases detection [8] or training machine learning models [9].

Nowadays, several tools can be used to squiggle visualization and basic processing. Squigualiser [10] enables visual inspection of squiggles with their basecalled sequences but does not support query-based signal extraction. MANASIG [11] or SquiggleKit [12] also allow signals visualization and, moreover, enable the export of signals for query sequences. However, their functionality is limited to the older nanopore data format. This presents an additional challenge, as the raw data formats have changed over time. Until 2023, the data were stored in the FAST5 format based on the HDF5 standard and organized signal and metadata hierarchically on a per-read basis. However, this format becomes inefficient for large datasets due to metadata overhead and slower parallel access, and in addition, due to frequent upgrades, it has become backwards incompatible. To address these limitations, ONT has introduced POD5, a new binary format designed for scalability, flat indexing, and high-throughput processing. However, many tools for raw signal processing are limited only to one nanopore format and thus cannot be used with either older or newer data.

Here, we introduce NANOSLAST – the **Nano**pore **S**ignal **L**ocal **A**lignment and **S**egmentation **T**ool to address these limitations. This lightweight Python-based application combines local sequence alignment (using BLAST) with mapping of basecalled reads to their corresponding raw signal segments. The tool supports both FAST5 and POD5 input formats, extracts and visualizes signal sections corresponding to user-defined sequences, and allows export of the resulting data in structured formats. The tool is suitable for a range of applications, including motif detection, signal benchmarking, and the preparation of training datasets for machine learning models in basecalling, methylation detection, or other signal-based analyses.

2. Materials and Methods

The proposed tool, NANOSLAST, is written in Python and designed for local analysis of raw signals generated by ONT sequencing devices. It provides a graphical user interface (GUI) for processing basecalled reads and their corresponding raw signal data and supports both nanopore data formats (FAST5, POD5). The application is organized into three main modules: Search, Export, and Plot, which are described below. Each module can be used separately or in combination to perform complete signal workflow from local sequence search to large-scale signal dataset export.

2.1. Supported file formats

The application works with raw nanopore data in either FAST5 or POD5 format. To connect the raw signals (squiggles) with basecalled nucleotides, the basecalled reads in SAM format must be provided, which can be generated by Dorado or Guppy (both ONT, UK). For specific tasks, BLAST [13] is used; thus, sequences in FASTA format are needed for creating the database, and for running the search, the query can be either a FASTA file or a simple sequence string. For signal export the tool supports CSV or Parquet formats.

2.2. Search module

The first module is the Search tab, which allows users to locate reads containing a specific sequence using local alignment via BLAST. For this purpose, the user must select the SAM file with basecalled reads, which is then converted to FASTA format. The obtained file is then used to create a nucleotide BLAST database using *makeblastdb* from the NCBI BLAST+ suite. Once the database is built, the user can search for a query sequence by *blastn*. The query sequence can be either pasted into the search box or uploaded as a FASTA file.

The BLAST output is saved as a tab-delimited TSV file, where each row corresponds to one hit. The table contains all necessary information such as read ID where the sequence was found and all BLAST parameters such as alignment score (bitscore), percentage of identical matches (pident), alignment length, start and end positions of the alignment or the expect value (evalue).

2.3. Plot module

The second module, the Plot tab, enables users to visualize the measured current signals with the basecalled nucleotides. The user must provide the SAM file and the folder with the raw nanopore data (in FAST5 or POD5 format). Then, by inserting a specific read ID into the entry field, the tool locates the corresponding signal, extracts the move information from the SAM file, maps the basecalled sequence onto the raw signal and plot the output.

The signal is shown as a line plot with nucleotide positions highlighted using colour-coded segments: adenine (A) in red, thymine (T) in blue, cytosine (C) in yellow, and guanine (G) in green. The x-axis represents time in milliseconds (based on sampling rate), and the y-axis shows current intensity in picoamperes.

2.4. Export module

The third module, the Export tab, allows users to extract raw signal segments that correspond to the query sequences identified by BLAST alignment. The user needs to provide the SAM file, the directory containing the raw nanopore data (FAST5 or POD5), and the TSV file generated by the BLAST search in the Search module.

Based on these inputs, the tool parses the SAM file and extracts the information about the move table, which indicates where nucleotides were basecalled, how many samples were trimmed from the signal and the basecaller sampling rate (stride). Using this information together with the alignment coordinates from the TSV file, the tool calculates the exact start and end positions of the signal segments.

The extracted signals are then exported as either a CSV or Parquet file. Each entry contains the read ID, the matched sequence, signal indices, orientation, and the raw signal values.

2.5. Implementation details

The NANOSLAST tool is implemented in Python and uses a modular design that allows easy maintenance and possible extension for further functions. The interface is built using the *customtkinter* library, which provides a modern and cross-platform GUI. Standard Python libraries are used to handle raw nanopore data formats, specifically, *h5py* for reading FAST5 files, *pod5* for accessing POD5 formats and *pandas* and *pyarrow* for data processing and their exporting in CSV or Parquet format. The signal visualization part is implemented utilizing the *matplotlib* library. The local alignment-based search is conducted by the *Bio.Blast* module from Biopython enables the NCBI BLAST+ tool to be used directly in Python. To keep the interface responsive, all time-consuming operations like BLAST search, reading files, signal extraction, or export run in background threads, which ensures that the application can be used normally, even during heavy processing.

The tool is lightweight and runs locally on the user's machine without the need for any specialized hardware. It can be used on a standard desktop or laptop and was tested on Windows 10 with Python 3.10.

3. Results and Discussion

The NANOSLAST tool provides a graphical interface that guides the user through the whole process from sequence searching to exporting signals. After selecting a sequence and completing the BLAST search, matching reads are displayed, and their corresponding raw signal segments are extracted and made available for visualization and export. Each step is done in a separate tab: *Search*, *Plot*, and *Export*, which are shown in Fig. 1A-C.

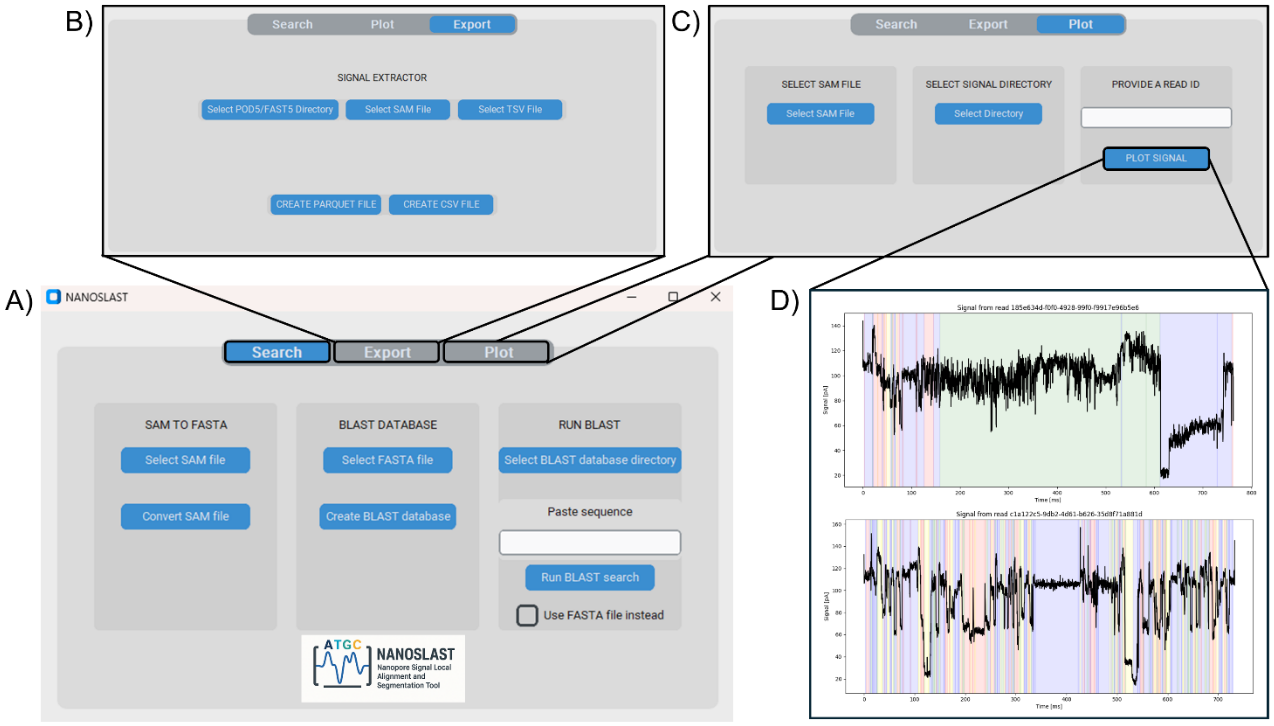

Fig. 1: Overview of the NANOSLAST graphical interface and output signal visualization: A) Search tab for locating query sequence using BLAST; B) Export tab for exporting signals corresponding to searched sequences; C) Plot tab for signal visualization; D) Example of two squiggles.

This signal visualization (see Fig. 1D) allows the user to study how the raw signal changes at individual bases and how each nucleotide influences the shape of the signal. It can help to see specific patterns, compare signal shapes between reads, or detect possible basecalling errors. The basecalled sequence is shown together with the signal trace, where each base is marked with a coloured segment, which makes it easy to interpret.

The main advantage of the tool is that it can directly link any sequence of interest to the exact part of the raw signal. By combining the alignment positions from the BLAST output with the move table and trimming information from the SAM file, the tool calculates the start and end of the signal segment that corresponds to the query sequence. This allows us to extract only the relevant part of the signal and use it for further analysis like motif search, machine learning training, or modified base detection. To our knowledge, no other existing tool provides this functionality in such a simple and interactive way, especially with support for both FAST5 and POD5 formats.

4. Conclusion

Working with raw nanopore signals is essential in cases where basecalling leads to information loss, such as modified base detection or training signal-based models. However, existing tools do not allow direct connection between a nucleotide sequence and its corresponding signal. To address this, we developed NANOSLAST, a lightweight Python tool with a graphical interface that combines local sequence alignment with raw signal extraction. It supports both FAST5 and POD5 formats providing an easy way to search, visualize, and export specific signal segments. The tool's source code is available on GitHub at: https://github.com/BioSys-BUT/NANOSLAST. NANOSLAST fills the gap between simple signal viewers and complex custom scripts, serving as a practical tool for researchers working with raw nanopore data.

Acknowledgements

This work was supported by GA23-05845S and NW24-09-00126.

References

[1] Y. Wang, Y. Zhao, A. Bollas, Y. Wang, and K. F. Au, "Nanopore sequencing technology, bioinformatics and applications," *Nat Biotechnol*, vol. 39, no. 11, pp. 1348–1365, Nov. 2021, doi: 10.1038/s41587-021-01108-x.

[2] D. Deamer, M. Akeson, and D. Branton, "Three decades of nanopore sequencing," *Nat Biotechnol*, vol. 34, no. 5, pp. 518–524, May 2016, doi: 10.1038/nbt.3423.

[3] F. J. Rang, W. P. Kloosterman, and J. de Ridder, "From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy," *Genome Biol*, vol. 19, no. 1, p. 90, Dec. 2018, doi: 10.1186/s13059-018-1462-9.

[4] H. Lu, F. Giordano, and Z. Ning, "Oxford Nanopore MinION Sequencing and Genome Assembly," *Genomics Proteomics Bioinformatics*, vol. 14, no. 5, pp. 265–279, Oct. 2016, doi: 10.1016/j.gpb.2016.05.004.

[5] N. D. Sanderson, K. M. V. Hopkins, M. Colpus, M. Parker, S. Lipworth, D. Crook, and N. Stoesser, "Evaluation of the accuracy of bacterial genome reconstruction with Oxford Nanopore R10.4.1 long-read-only sequencing," *Microb Genom*, vol. 10, no. 5, May 2024, doi: 10.1099/mgen.0.001246.

[6] W. Liu-Wei, W. van der Toorn, P. Bohn, M. Hölzer, R. P. Smyth, and M. von Kleist, "Sequencing accuracy and systematic errors of nanopore direct RNA sequencing," *BMC Genomics*, vol. 25, no. 1, p. 528, May 2024, doi: 10.1186/s12864-024-10440-w.

[7] M. Nykrynova, M. Bezdicek, M. Lengerova, and H. Skutkova, "Bacterial phenotype prediction based on methylation site profiles," in *2023 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, IEEE, Aug. 2023, pp. 1–6. doi: 10.1109/CIBCB56990.2023.10264900.

[8] M. U. Ahsan, A. Gouru, J. Chan, W. Zhou, and K. Wang, "A signal processing and deep learning framework for methylation detection using Oxford Nanopore sequencing," *Nat Commun*, vol. 15, no. 1, p. 1448, Feb. 2024, doi: 10.1038/s41467-024-45778-y.

[9] M. Nykrynova, R. Jakubicek, V. Barton, M. Bezdicek, M. Lengerova, and H. Skutkova, "Using deep learning for gene detection and classification in raw nanopore signals," *Front Microbiol*, vol. 13, Sep. 2022, doi: 10.3389/fmicb.2022.942179.

[10] H. Samarakoon, K. Liyanage, J. M. Ferguson, S. Parameswaran, H. Gamaarachchi, and I. W. Deveson, "Interactive visualization of nanopore sequencing signal data with *Squigualiser*," *Bioinformatics*, vol. 40, no. 8, Aug. 2024, doi: 10.1093/bioinformatics/btae501.

[11] V. Barton, M. Nykrynova, and H. Skutkova, "MANASIG: Python Package to Manipulate Nanopore Signals From Sequencing Files," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, IEEE, Dec. 2021, pp. 1941–1947. doi: 10.1109/BIBM52615.2021.9669821.

[12] J. M. Ferguson and M. A. Smith, "SquiggleKit: a toolkit for manipulating nanopore signal data," *Bioinformatics*, vol. 35, no. 24, pp. 5372–5373, Dec. 2019, doi: 10.1093/bioinformatics/btz586.

[13] C. Camacho, G. Coulouris, V. Avagyan, N. Ma, J. Papadopoulos, K. Bealer, and T. L. Madden, "BLAST+: architecture and applications," *BMC Bioinformatics*, vol. 10, no. 1, p. 421, Dec. 2009, doi: 10.1186/1471-2105-10-421.