Technical Note

Rescuing the "Unclassified": An Automated Pipeline for Maximizing Sequence Yield in Multiplexed Nanopore 16S rRNA Sequencing

Adrian Ionascu * and Nicoleta-Denisa Constantin

Posted Date: 27 March 2026

doi: 10.20944/preprints202603.2250.v1

Keywords: ONT; 16S rRNA sequencing; bioinformatics; recovery unclassified reads

 doi:10.20944/preprints202603.2250.v1

Technical Note

Rescuing the "Unclassified": An Automated Pipeline for Maximizing Sequence Yield in Multiplexed Nanopore 16S rRNA Sequencing

Adrian Ionascu * and Nicoleta-Denisa Constantin

Department of Genetics, Faculty of Biology, University of Bucharest, Romania, 060101
* Correspondence: a.ionascu20@s.bio.unibuc.ro

Abstract

Multiplexed 16S rRNA gene sequencing using Oxford Nanopore Technologies frequently results in a significant proportion of reads being categorized as "unclassified" by the standard *Dorado* demultiplexer. We present a fully automated Bash-based pipeline designed to recover and accurately re-attribute these unclassified reads to their corresponding barcode. By implementing iterative sequence-based alignment via BLASTn and size-filtering constraints, our tool improves data utilization and taxonomic depth. Preliminary testing demonstrates a robust recovery of previously discarded sequences, providing a cost-effective solution for improving the resolution of microbiome studies.

Keywords: ONT; 16S rRNA sequencing; bioinformatics; recovery unclassified reads

1. Introduction

Oxford Nanopore Technologies (ONT) has revolutionized microbial genomics by enabling long-read, real-time sequencing of the full ~1,500 bp 16S rRNA gene. This approach provides superior taxonomic resolution compared to short-read platforms that target isolated hypervariable regions [1]. Despite these advantages, demultiplexing remains a critical bottleneck. Standard demultiplexers, such as *Dorado*, often fail to assign reads when the barcode sequence is obscured by the characteristic "noise" or signal degradation common at the beginning and end of nanopore reads [2].

2. Materials and Methods

The recovery script is a modular Bash-based pipeline that integrates *ncbi-blast+* [3] *SeqKit* [4], and *NanoPlot* [5]. To ensure biological integrity, the script applies a strict size-selection filter. Sequences falling outside the user-defined range (e.g., <1000 bp or >2000 bp for 16S amplicons) are segregated into dedicated "fail" directories. This ensures that only full-length or near-full-length 16S reads proceed to the alignment phase.

The core recovery mechanism utilizes BLASTn to align these filtered sequences against a database of known barcodes. By adjusting the *word_size* and *percent_identity* parameters, the script can identify barcode sequences even when they contain the insertions or deletions (indels) typical of nanopore chemistry. Once a unique hit is confirmed, the script extracts the corresponding raw read record from the original pool, preserving the associated quality scores for downstream taxonomic classification or phylogenetic analysis.

The script is available as open-access on GitHub at https://github.com/A-Ionascu/ONT-RUR-16S. Our implementation was created using Google Gemini [6].

 doi:10.20944/preprints202603.2250.v1

3. Results

3.1. Reporting

The pipeline produces a structured output that facilitates immediate data verification and reporting. The primary analytical result is a recovery report, titled *recovery_statistics.csv*. This document calculates the exact contribution of the recovery pipeline by comparing the original demultiplexed read counts with the corresponding recovered reads. The script provides a recovery percentage, in order to evaluate how much additional data was gained through the pipeline.

In addition to quantitative metrics, the script provides a qualitative assessment of the recovered data through automated *NanoPlot* [5] integration. For every barcode identified in the unclassified pool, a detailed quality profile is generated, including read length histograms and quality score distributions. To streamline large-scale experiments, the pipeline further utilizes an integrated R-script that scrapes individual *NanoPlot* [5] statistics into a summary table. This allows for a direct comparison of the quality profiles of recovered reads against the original demultiplexed reads, ensuring that the recovered sequences meet the standards required for microbial profiling.

3.2. Usage

3.2.1. Functional Framework: Compulsory Arguments

The operational integrity of the pipeline is predicated on four primary compulsory arguments that define the search space and the final data architecture. The *--input* argument is used for providing the raw pool of unclassified FASTQ sequences. The *--barcodes* flag denotes the target molecular signatures for alignment. The package contains a file with all ONT 16S barcodes available for SQK-16S024 kit. The *--reads_dir* argument directs the script to the existing demultiplexed data generated by standard ONT software (e.g., Dorado or Guppy). The implication of this requirement is that the script does not function as a mere isolated aligner, but as a holistic integration tool. By referencing the original classified reads, the pipeline concatenates newly recovered sequences with their original counterparts to produce a unified dataset. This connectivity ensures that the final output is a complete representation of the sample's total microbial diversity, maintaining the biological continuity of the experiment. The *--output* argument allows for the creating a user-defined output directory.

3.2.2. Precision Tuning: Optional Arguments and Bioinformatic Control

While the compulsory arguments set the structural framework, the optional arguments act as precision tuning to balance bioinformatic stringency with computational performance. The *--word_size* and *--perc_identity* parameters allow to modulate the sensitivity of the BLASTn algorithm. The implementation of *--min_length* and *--max_length* filters provides a biological safeguard by constraining the recovery process to the expected dimensions of the 16S rRNA gene (~1,500 bp). These arguments prevent the erroneous inclusion of non-target artifact fragments that could otherwise skew diversity indices. The *--nanoplot* option enables resource management, allowing users to bypass intensive visualization steps for large concatenated files when only raw sequence recovery is required. Similarly, the *--threads* argument, a standard implementation in Bash programming, sets the number of computational threads for running the script. Together, these optional flags empower the user to adapt the pipeline to the specific quality profile and biological context of their unique sequencing run, maximizing data salvage without sacrificing sequence integrity.

3.2.3. User Manual

DESCRIPTION:

Automated pipeline for Nanopore 16S unclassified barcode recovery, BLAST validation, and NanoPlot quality control.

REQUIRED ARGUMENTS:

-i,	--input	<FILE>	Unclassified FASTQ file.
-o,	--output	<DIR>	Directory where results will be saved.

 doi:10.20944/preprints202603.2250.v1


```
    -b,  --barcodes       <FILE>  FASTA file containing the barcode sequences.
    -r,  --reads_dir      <DIR>   Folder containing demultiplexed FASTQ files.
    OPTIONAL ARGUMENTS:
    -w,  --word_size      <INT>   Word size for BLASTn (barcode length = 24) [Default: 11].
    -p,  --perc_identity  <INT>   Percent identity for BLASTn alignment (0-100) [Default: 0].
    -min, --min_length    <INT>   Minimum read length to keep [Default: 0].
    -max, --max_length    <INT>   Maximum read length to keep [Default: None].
    -nanoplot, --nanoplot_concatenated  <on|off>  Run NanoPlot for concatenated reads [Default: on].
    -t,  --threads        <INT>   Number of CPU threads to use [Default: 4].
    -h,  --help                   Display this help manual and exit.
```

4. Discussion

In microbiome research, rare species, those representing less than 0.1% of a community, are often the first to disappear when sequencing depth is insufficient. By recovering a portion of unclassified reads, this script directly increases the sensitivity of the assay, allowing for the detection of rare pathogens or low-abundance environmental microbes that would otherwise go unnoticed. This is particularly relevant in clinical diagnostics, where the failure to classify several reads could result in a missed identification of a slow-growing or low-titer infectious agent.

Sequencing flow cells and reagents are significant expenses in any genomic study. Standard workflows that discard unclassified reads essentially waste a portion of the flow cell's capacity. By implementing an automated recovery step, researchers maximize the value of every run, effectively lowering the "cost-per-read" and allowing for more ambitious study designs on limited budgets.

Beyond cost, the script promotes a higher standard of data transparency. Because it organizes filtered-out reads into specific files (*lower_length_fail* and *upper_length_fail*), users can investigate why reads were unclassified, whether due to library fragmentation, adapter dimers, or poor basecalling.

Author Contributions: Conceptualization, A.I. and N.D.C.; methodology, A.I. and N.D.C.; software, A.I. and N.D.C.; validation, A.I. and N.D.C.; formal analysis, A.I. and N.D.C.; investigation, A.I. and N.D.C.; resources, A.I. and N.D.C.; data curation, A.I. and N.D.C.; writing—original draft preparation, A.I. and N.D.C.; writing—review and editing, A.I. and N.D.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The script is available as open-access on GitHub at https://github.com/A-Ionascu/ONT-RUR-16S.

Acknowledgments: We acknowledge the members of the TRANSCEND project (Grant No. PN-III-P4-PCE-2021-1797) for providing the experimental sequencing data used in the development and validation of this software.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Esberg, A.; Fries, N.; Haworth, S.; Johansson, I. Saliva microbiome profiling by full-gene 16S rRNA Oxford Nanopore Technology versus Illumina MiSeq sequencing. *npj Biofilms Microbiomes* **2024**, *10*, 149.
2. Brunet, S.; Grankvist, A.; Jaen-Luchoro, D.; Bergdahl, M.; Tison, J.L.; Wester, A.; Elfving, K.; Brandenburg, J.; Gullsby, K.; Lindsten, C.; Arvidsson, L.O.; Larsson, H.; Eilers, H.; Söderlund Strand, A.; Lannefors, M.; Keskitalo, J.; Rylander, F.; Welander, J.; Bergman Jungestrom, M.; Geörg, M.; Kaden, R.; Karlsson, I.; Linde,

 doi:10.20944/preprints202603.2250.v1

A.M.; Mernelius, S.; Berglind, L.; Feuk, L.; Kerje, S.; Karlsson, L.; Sjödin, A.; Guerra-Blomqvist, L.; Wallin, F.; Fagerström, A.; Vondracek, M.; Mölling, P.; Tång Hallbäck, E. Nationwide multicentre study of Nanopore long-read sequencing for 16S rRNA-species identification. *Eur. J. Clin. Microbiol. Infect. Dis.* **2025**, *44*, 1907-1916.

3. Camacho, C.; Coulouris, G.; Avagyan, V.; Ma, N.; Papadopoulos, J.; Bealer, K.; Madden, T.L. BLAST+: architecture and applications. *BMC Bioinform.* **2009**, *10*, 421.
4. Shen, W.; Le, S.; Li, Y.; Hu, F. SeqKit: a cross-platform and ultrafast toolkit for FASTA/Q file manipulation. *PLoS One* **2016**, *11*, e0163962.
5. De Coster, W.; D'hert, S.; Schultz, D.T.; Cruts, M.; Van Broeckhoven, C. NanoPack: visualizing and processing long-read sequencing data. *Bioinformatics* **2018**, *34*, 2666-2669.
6. Google. Gemini 3.1 Pro Large Language Model. https://gemini.google.com **2026**.