Fishnet simplifies and accelerates signal-to-sequence alignment in Nanopore sequencing

Vincent Dietrich
University Medical Center of the Johannes Gutenberg University Mainz

Lioba Lehmann
University Medical Center of the Johannes Gutenberg University Mainz

Stefan Pastore
University Medical Center of the Johannes Gutenberg University Mainz

Stefan Mündnich
Johannes Gutenberg University Mainz

Lukas Walz
Johannes Gutenberg University Mainz

Mark Helm
Johannes Gutenberg University Mainz

Kristina Friedland
Johannes Gutenberg University Mainz

Susanne Gerber
sugerber@uni-mainz.de

University Medical Center of the Johannes Gutenberg University Mainz

software

Keywords: RNA modification, Nanopore sequencing, Signal-to-sequence alignment, Resquiggling, Benchmarking, Benchmarking Signal-to-sequence alignment, Rust implementation Remora

Posted Date: January 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-8345719/v1

Additional Declarations: No competing interests reported.

Abstract

Nanopore sequencing preserves native DNA and RNA modifications and encodes them directly in electrical signal, but extracting this information requires accurate signal-to-sequence alignment. Existing tools perform this reliably yet often demand metadata handling or format conversion. We present Fishnet, a lightweight and fast aligner that reimplements the Remora alignment algorithm while removing surrounding overhead. Fishnet produces near-identical alignments more than thirty times faster and provides a simple command-line interface for alignment and downstream formatting. Benchmarks demonstrate high concordance between alignment tools and Fishnet's mostly superior speed. Analyses of synthetic RNA constructs prove its practical utility for streamlined studies of modified nucleotides.

Background

Chemical modifications to nucleic acids introduce large structural and functional variability to genomes and transcriptomes. The modification landscape in RNA is especially diverse, with more than 170 distinct types of modifications (Boccaletto et al. 2022). They are dynamic and occur in coding and non-coding RNA, where they assume diverse, primarily regulatory roles (Lee et al. 2020; Lesbirel and Wilson 2019; Mao et al. 2019; Yu et al. 2018; Hewel et al. 2025). For example, N1-methyladenosine (m^1A) occurs abundantly in both cytoplasmic and mitochondrial tRNA, where it is involved in the tightly regulated mitochondrial electron transport chain, including known occurrences in the COX1, COX2 and COX3 subunits of the cytochrome c oxidase (Mayr et al. 2015; Smits et al. 2010). As such, its dysregulation is implicated in cancer, Alzheimer's disease (Jörg et al. 2024; Xiong et al. 2023) and multiple mitochondrial disorders (Richter et al. 2018; Shafik et al. 2022; Zhang and Jia 2018). These observations highlight the functional and disease-related complexity of RNA modifications and reinforce the need for accurate characterization of modification-dependent mechanisms.

A better understanding of these processes is gained by accurate single-nucleotide mapping and quantification of modified residues across genomes and transcriptomes. Next-generation sequencing (NGS) utilizes indirect detection approaches for mapping modified nucleotides. These rely on modification-specific chemicals that induce characteristic error or cutoff patterns at modified positions (Zhang et al. 2022; Spangenberg et al. 2025; Hewel et al. 2025). While these methods are accurate, they are modification-specific and require specific treatment prior to sequencing (X. Chen et al. 2025). In contrast, Nanopore sequencing as established by Oxford Nanopore Technologies (ONT), allows for sequencing native DNA or RNA molecules. This enables direct detection of modified nucleotides through modification-induced pattern shifts in the measured current signals (Furlan et al. 2021; White and Hesselberth 2022; Xu and Seki 2020; Diensthuber and Novoa 2025). While these characteristics can in principle be exploited for all types of modifications, the pattern shifts are usually subtle (Cruciani and Novoa 2025), and often rely on deep-learning models for accurate identification. ONT provides modification calling for a handful of DNA and RNA modification types in their basecalling software Dorado[1]. Beyond the officially supported modification calling, the community continuously expands existing options with custom tools and models (Alagna et al. 2025; Wu et al. 2025; Pagès-Gallego et al. 2025; Li et al. 2025; Vujaklija et al. 2025; Rübsam et al. 2025; H.-X. Chen et al. 2025).

An important and commonly performed step when working with Nanopore sequencing data on the signal level is the alignment of the signal to the corresponding base sequence. This can be the basecalled (*query*) sequence or a reference sequence. In this signal-to-sequence alignment, or *resquiggling*, a chunk of signal measurements gets assigned to a single base, providing a base-centered view on the measured signal in the process (see Fig. 1). Resquiggling typically relies on dynamic time warping and requires loading large signal datasets, making it computationally demanding. Nonetheless, it is an essential step in gaining a better understanding of how a given sequence context influences the signal, and how modified residues change it. Accurate signal-to-sequence alignment is therefore essential for modification calling and training machine-learning models on Nanopore signals.

Multiple tools include functionalities to generate signal-to-sequence alignments from raw sequencing data and corresponding basecalled or reference-mapped sequences. F5c reimplements functionality from the earlier, no longer maintained Nanopolish (Liu et al. 2021) with added support for recent sequencing chemistries and GPU-acceleration (Gamaarachchi et al. 2020), but it requires manual preprocessing and file conversions. Uncalled4 provides extensive postprocessing and visualization features in addition to alignment (Kovaka et al. 2025), offering a modern end-to-end solution. However, it requires explicit specification of flowcell-types and sequencing kits, adding configuration overhead. Currently developed by ONT, Remora[2] provides signal-to-sequence alignment functions for the latest chemistries and file formats, but it does not provide a proper command line interface, instead requiring users to integrate the existing functions into custom Python scripts without common convenience features. While these tools enable signal-to-sequence alignment, the functionalities often require additional preprocessing, come with many additional features that are not directly relevant for the alignment process itself and entail – to varying degrees – long processing times.

To streamline this process, we developed Fishnet, a lightweight, accessible, fast, and alignment-centered signal-to-sequence aligner for both DNA and RNA. Designed with multi-threading in mind, it outperforms established tools in the majority of tested cases, with > 30-fold speedups over Remora. Fishnet reimplements the alignment algorithm that is used in Remora, producing alignments with only minuscule differences and thus enabling its usage as a direct stand-in for analysis pipelines relying on Remora-based signal-to-sequence alignments.

Beyond the alignment itself, Fishnet provides downstream reformatting and filtering functionalities that allow straightforward preparation for targeted exploratory analyses or machine learning processing. All functions are available via a minimal command line interface, where the *align* command handles the signal-to-sequence alignment and the *reformat* command handles further processing options. Fishnet runs natively on any Linux distribution (both x64 and arm64) and Windows and can be accessed directly from a single executable without any installation needed. It is openly available from the GitHub repository (https://github.com/dietvin/fishnet). Here, we benchmark Fishnet's processing speed and systematically compare produced alignments against established tools. We demonstrate its utility, through the analysis of individual m^{1}A sites in synthetic *COX1*, *COX2* and *COX3* RNA oligos.

Results

Comparing processing speed

The processing times were systematically measured for Fishnet, Remora, Uncalled4 and f5c with varying number of parallel threads using subsets of ONT's *Genome in a Bottle* DNA dataset. The data was split into subsets containing short, medium and long reads exclusively. For each read length, subsets of 100, 1000, 10000 and 100000 reads were extracted, and both query- and reference-to-signal alignments were calculated with all tool-setting-dataset combinations (see Methods, Comparing processing times).

The processing times for query-to-signal alignments on long reads showed an exponential runtime growth with an increasing number of reads. Overall, Fishnet demonstrated the fastest performance followed by f5c, Uncalled4 and lastly Remora. The alignment of 100000 long reads to their query sequences took Fishnet 30:58 minutes, while f5c needed 1:00:00 hour, Uncalled4 2:30:19 hours and Remora 16:50:34 hours (Fig. 2a).

The increase in processing time was stable for all tools except Uncalled4, which started out as the slowest but showed a notable speedup between 1000 and 10000 reads. This was reflected in the speedup of Fishnet over Uncalled4, which Fishnet was approximately 37-fold with 100 reads, 26-fold with 1000 reads, and 7- and 5-fold with 10000 and 100000 reads, respectively. The speedup of Fishnet over f5c was more stable, increasing slightly from 1.4-fold for 100 reads to 1.9-fold for 100000 reads. The largest and best-scaling speedup occurred over Remora, which started out at roughly 22-fold and increased to 33-fold from 100 to 100000 reads (Fig. 2b).

Overall processing times decreased over the board when processing medium and short read lengths. For 100000 reads for example, the processing times decreased to 14:56 minutes and 5:44 minutes for Fishnet, 21:50 minutes and 5:03 minutes for f5c, 43:46 minutes and 9:30 minutes for Uncalled4 and 6:59:37 hours and 2:16:41 hours for Remora, when aligning medium and short reads, respectively (Supplementary Fig. 1a & c). Notably, only with short reads, f5c outperformed Fishnet by 41 seconds.

The speedup of Fishnet over the other tools was largely similar between medium and long reads. With an increasing number of medium reads, Fishnet performed slightly better than f5c and notably better than Remora, while Uncalled4 scaled more efficiently when including more reads (Supplementary Fig. 1b). When aligning short reads (Supplementary Fig. 1d), while f5c was slightly faster, the speedup peaked with 1000 and 10000 reads at approximately two times faster. With 100000 reads, f5c outperformed Fishnet by a factor of 1.1-fold. The speedup of Fishnet over Remora started at 61-fold for 100 short reads but dropped to more consistent factors between 29-fold for 10000 reads and 24-fold for 100000 reads. The speedup of Fishnet over Uncalled4 behaved similarly to medium and long reads, although the maximum speedup was lower at 11-fold.

For reference-to-signal alignments, the tools also showed an exponential increase in processing time with an increasing amount of data. For 100000 long reads, Fishnet again measured the fastest time with 32:32 minutes, followed by Uncalled4 with 2:21:37 hours, f5c with 5:13:51 hours and Remora with 16:46:59 hours (Fig. 2c). Compared to the query-to-signal alignment, the reference-to-signal alignment took Fishnet 1:33 minutes longer. For f5c the difference was more pronounced with the reference-to-signal alignment running 4:12:51 hours longer. For Uncalled4 and Remora, the reference-to-signal alignment ran slower as well, with a difference of 8:42 minutes and 3:35 minutes, respectively.

Similar to the query-to-signal alignment, the processing time scaled more efficiently for Uncalled4 between 1000 and 10000 long reads (Fig. 2d), with Fishnet achieving 23-fold speedups with 1000 reads and 6-fold with 10000 reads. With 100000 reads, Fishnet sped up 4-fold over Uncalled4. With an increasing number of long reads, the speedup of Fishnet over f5c increased from 5-fold with 100 reads to almost 10-fold with 100000 reads. The speedup of Fishnet over Remora started at approximately 20-fold with 100 long reads and increased steadily to 31-fold for 100000 reads. Like before, processing times also scaled with the read lengths, for 100000 medium and short reads, respectively, dropping to 15:41 and 5:45 minutes for Fishnet, 41:42 and 9:35 minutes for Uncalled4, 2:07:55 hours and 23:50 minutes for f5c, and 6:57:48 and 2:12:52 for Remora (Supplementary Fig. 2a & c).

For reference-to-signal alignments, the speedup of Fishnet over the other tools behaved largely the same for medium reads compared to long reads, although the speedup overall was slightly lower, with Fishnet measuring speedup of 2.6-fold over Uncalled4, 8.1-fold over f5c and 27-fold over Remora for 100000 reads (Supplementary Fig. 2b). The overall changes in speedup factors with different amounts of reads are largely consistent between medium and long reads. With short reads, the speedup factors are largely consistent as well, with the exception that the speedup of Fishnet over f5c peaks with 10000 reads at 6.4-fold (Supplementary Fig. 2d).

Comparing produced alignments

To compare alignments, 100000 medium-length reads were aligned to the query sequences using Fishnet, Remora, Uncalled4 and f5c. Differences between alignments of the same reads were quantified using the normalized mean distance (NMD) between two alignments. Reference-to-signal alignments were processed in the same way, but due to limitations of the *eventalign* output format generated by Uncalled4 and f5c, these were only compared between Fishnet and Remora (see Methods, Comparing alignments).

The query-to-signal alignments showed high similarity for all pairwise comparisons of the four tools. Here, all NMD distributions mostly aggregate between 0 and 0.0003, which corresponds to an average difference in the signal boundaries of 0.03% of the signal length. Only a small fraction of compared reads showed larger deviation in the alignments. As such, the distributions mainly consisted of a flat, but wide chunk after an initial peak at 0 (Fig. 3).

Alignments of Fishnet and Remora showed the highest degree of similarity, with the median NMD at 0.000069%, and the 95th percentile at 0.002% of the signal length. The largest NMD score is approximately 0.13% of the signal length (Fig. 3b). Uncalled4 and f5c showed a notable increase in NMD scores, with the median at 0.001% and the 95th percentile at 0.04% of the signal length. Here, the largest NMD was at 18.4% of the signal length (Fig. 3d).

The differences in NMD scores further increased when comparing Fishnet to f5c and Uncalled4, with median NMD scores of approximately 0.01% of the signal length for both comparisons. The 95th percentiles of the NMDs were at 0.06% and 0.02%, respectively. Only the maximum NMD showed notable differences, with 39% for the Fishnet-f5c (Fig. 3b), and 2% for the Fishnet-Uncalled4 comparison (Fig. 3c).

In line with the high similarities between Fishnet and Remora, comparisons of Remora with f5c and Uncalled4 resulted in virtually the same results observed in the same comparisons with Fishnet (Fig. 3e, f).

The query-to-signal alignment for read *0003f68c-1572-46b5-af5b-c186bb0482ed* reflected the trends observed in the NMD scores between the tools. Fishnet and Remora produced identical alignment paths with no discernible differences. The f5c and Uncalled4 alignments closely matched throughout most of the read, with minor deviations at the start that converged to perfect overlap by approximately base 40. The primary difference between these tools emerged at the alignment terminus, where Uncalled4 extended substantially beyond the signal length (Fig. 3g). When comparing the perfectly concordant Fishnet-Remora pair to the largely concordant f5c-Uncalled4 pair, subtle alignment differences appeared throughout the entire read, including visible boundary shifts at the beginning, middle and end.

The comparison of the reference-to-signal alignments between Fishnet and Remora was analogous to the one of the query-to-signal alignments, with a median NMD at 0.00007% of the signal length. Here, the 95th percentile was 0.0002% and the maximum NMD at 0.2% of the signal length (Fig. 3h). The example reference-to-signal alignment of read *0003f68c-1572-46b5-af5b-c186bb0482ed* also shows no discernable deviations between the alignments (Fig. 3i).

Analysis of an m^1A site in RNA constructs

Fishnet was used to process Nanopore direct RNA sequencing data from constructs containing a single m^1A site in the center and corresponding unmodified control constructs, which allowed for a direct comparison between modified and unmodified signals. Three different contexts were processed, labelled *COX1*, *COX2* and *COX3*. After aligning signals to the reference sequences using Fishnet's *align* command, the signal chunks were reformatted by calculating representative statistics for each base. These statistics allowed for a statistical comparison between modified and unmodified reads, where effect sizes between the two conditions were estimated using Cohen's d (Cohen 2013; 1992). Additionally, aligned signal chunks were interpolated into a uniform shape for all bases, which allowed for dimensionality reduction using UMAP. The interpolation and subsequent UMAP were performed using the interpolated signal and dwell times from only the central modification site, as well as including one, two, and four bases up- and downstream from it. All reformatting steps were performed using Fishnet's *reformat* command (see Methods, Analysis of m¹A sites in RNA constructs).

For *COX1*, the signals from modified reads showed a notable spread towards lower signal intensities at the – 5 to + 2 sites relative to the central m^1A site compared to the signals from unmodified reads (Fig. 4a). These deviations were less pronounced in the *COX2* sample, where only slight deviations towards higher intensities occurred at the – 2 and + 1, in addition to the central m^1A site itself, between modified and unmodified data (Fig. 4b). Similarly to *COX2*, the signals in the *COX3* sample showed only subtle differences, with tendencies towards higher intensities at the – 4, -2 and + 1 sites with m^1A present (Fig. 4c).

Considering all 42 bases up- and downstream from the central m^1A site showed that an increase in the spread of signal intensities from modified reads is only visible in the surroundings of the modification site, with a slight shift towards downstream bases. With increasing distance, both up- and downstream from this central region, signals from modified bases did not show such variation in comparison, with clear differences between modified and unmodified signals only occurring once the reference sequences of the constructs differ from each other. At these outmost positions, there were clear separations between modified and unmodified reads. This was the case for *COX1* (Supplementary Fig. 5a), *COX2* (Supplementary Fig. 6a) and *COX3* (Supplementary Fig. 7a).

The base-wise distributions of mean signal intensities matched the signal curves for all three samples. For *COX1*, the distributions tended to spread further towards smaller intensities, most notably in the range from the – 5 to the + 1 site relative to the central modification site. For all these seven bases, the median of the mean intensities from modified reads were below the 25th percentile of the unmodified reads (Fig. 4d, top). The effect sizes underline the extent of the differences, with scores below d=-0.5 for all bases in this region. Notably, from the – 5 to -2 site, the effect sizes ranged between d=-0.85 and d=-0.88, and the largest difference occurred at the modification site itself, with d=-0.95 (Fig. 4d, bottom). Beyond the direct surroundings of the modification site, no other intermediate regions show comparable deviations over multiple bases. Differences only increase once close to the outmost bases where the sequences between modified and unmodified reads differ (Supplementary Fig. 5b).

In the *COX2* signals, the mean intensity differed to a lesser extent around the central site between modified and unmodified reads. The most distinct differences between occurred at the modification site, with a more wide-spread distribution that is shifted towards lower intensities in the modified data. At surrounding sites, while the 5th and 95th percentiles spread further in the modified reads, the differences are less pronounced regarding the 25th and 75th percentiles (Fig. 4e, top). Accordingly, the effect sizes are not as distinct in the latter. On the modification site and the – 4 site effect sizes are smaller than d=-0.5 (-0.75 and – 0.6) (Fig. 4e, bottom). Like with *COX1*, the deviations only increase at the bases close to the differing sequences (Supplementary Fig. 6b).

The *COX3* data shows, similar to *COX2*, that the mean signal intensity changes foremost in the 5th and 95th percentiles around the modification site, with more spread in the modified data. The 25th and 75th percentiles indicate a slight shift towards lower intensities in modified reads, though the differences are subtle for most bases (Fig. 4f, top). Only the modification site itself shows a clear shift towards lower intensities with an effect size of d=-0.8 (Fig. 4f, bottom). Beyond the direct surroundings, the differences do not increase between modified and unmodified data (Supplementary Fig. 7b).

Dwell times were consistent for the most part throughout the three samples, with the standardized dwell times aggregating in the range from − 0.5 to 0 for all samples. Notable exceptions for *COX1* were sites + 1 and − 9 relative to the central modification site, where the dwell time in modified reads is larger in comparison, effect sizes of d = 0.52 and d = 0.54, respectively (Fig. 4g). The highest divergence between modified and unmodified data occurred at the − 2 and − 3 sites, where modified dwell times decrease in comparison, reaching effect sizes of -1.2 and − 0.7, respectively. These two sites were the same ones that already showed a notable decrease in the signal intensity, as described above.

For *COX2*, the dwell times diverged to a slightly higher degree between the − 7 and + 1 sites compared to the remaining surrounding bases, though the direction was not consistent (Fig. 4h). Sites − 7, -5, -3, -2 and + 1 showed subtle increases in dwell time in the modified data, reaching effect sizes that do not exceed d = 0.46. The remaining sites in the region showed a more notable decrease in dwell time in the modified data, with effect sizes of d=-0.5 at the − 5 site, d=-0.76 at the − 4 site, and d=-0.84. As with *COX1*, the largest decrease in dwell time coincides with the largest decrease in signal intensity.

The most notable differences in the *COX3* dwell times occurred at the − 4 site with an increase in dwell times in modified data, and a more distinct decrease in dwell times at the − 1 site (Fig. 4i, top). The two sites showed effect sizes of d = 0.57 and d=-1, respectively (Fig. 4i, bottom). Consistent with the other samples, the − 1 site corresponds to the site with the largest decrease in signal intensity.

Beyond the mentioned positions, the different samples do not show notable differences in the regions further away from the modification site, with only individual bases near the outmost bases showing in- or decreased dwell times (Supplementary Fig. 5c, 6c and 7c).

In summary, the direct signal intensities, the aggregated mean signal intensities and dwell times all show a high similarity throughout the regions flanking the center of the sequences between the *COX1*, *COX2* and *COX3* sample. Notable differences occur only in the center region, where the sequences differ between the three samples (compare Supplementary Fig. 5, Supplementary Fig. 6 and Supplementary Fig. 7).

For *COX1*, performing UMAP on the interpolated signal data and dwell times from the central (modified) base and one base up- and downstream (three bases in total) from it, resulted in two larger clusters in the top-left and center-left consisting of mostly modified reads with little overlap with unmodified ones. Additionally, modified reads showed an aggregation in the center-right regions, though with a slightly higher overlap with unmodified reads. Unmodified reads were most densely accumulated in the center region. All other regions showed a high overlap between modified and unmodified reads (Fig. 4j). Decreasing the amount of information provided to the UMAP to only the central base resulted in no clear separation between modified and unmodified reads (Supplementary Fig. 8a). Increasing the amount of information with 2 and 4 surrounding bases (five and nine bases in total, respectively) resulted in largely similar clustering of the data with more separate groups of modified reads and a central aggregation of unmodified ones, although the number of small groups increases with more information given (Supplementary Fig. 8b, c).

The UMAP of the three bases *COX2* data showed a high degree of overlap between modified and unmodified reads compared to *COX1*. The cleanest separation of modified reads occurred in the bottom-center to bottom-left region, while unmodified reads group more towards the center (Fig. 4k). Decreasing the number of bases for the UMAP to only the central one decreased the amount of overlap even more (Supplementary Fig. 8d). With five bases included, one subgroup of modified reads separated clearly (Supplementary Fig. 8e), and with nine bases a second smaller modified-exclusive cluster appeared (Supplementary Fig. 8f).

Most of the *COX3* reads group in one large cluster when using three bases as input. In this large cluster, modified reads tend to group towards the left edge of the cluster, while unmodified reads group more on the right edge. Additionally, a separate group of modified reads cluster in the center-right region (Fig. 4l). As with the previous samples, providing only the information of the center base results in worse separation (Supplementary Fig. 8g). With five bases provided, the large cluster observed with three bases was still visible, though more fractured (Supplementary Fig. 8h). This fracturing increases with nine bases included (Supplementary Fig. 8i).

Discussion

In this work we present Fishnet, a lightweight and fast signal-to-sequence aligner for Nanopore sequencing data that reimplements and extends the algorithm used in Oxford Nanopore Technologies (ONT) Remora tool. Fishnet aims to provide a streamlined, accessible and high-throughput alternative while maintaining compatibility with Remora-alignment based downstream workflows.

Computational performance

We benchmarked Fishnet against three established tools – f5c, Uncalled4 and Remora – across both query-to-signal and reference-to-signal alignment tasks. Compared to f5c, Fishnet showed substantially better scaling efficiency for reference-to-signal alignments, achieving 5- to 10-fold speedups from 100 to 100000 long reads. Query-to-signal speedup were more modest (~ 2-fold). Notably, f5c was specifically designed for GPU acceleration, and was tested here only in CPU mode due to hardware constraints. Reportedly, GPU acceleration provides approximately 3-4.5-fold speedup compared to CPU-only execution (Gamaarachchi et al. 2020). Given this speedup, Fishnet would likely maintain a speed advantage over GPU-accelerated f5c for reference-to-signal alignments, while GPU-accelerated query-to-signal alignments would likely match or exceed Fishnet’s performance.

Fishnet outperformed Uncalled4 across all tested scenarios, with up to a 37-fold speedup for small datasets and a 5-fold speedup for large ones. The diminishing margin at higher read counts indicates that Uncalled4 exhibits better scaling characteristics at larger dataset sizes. This convergence pattern suggests that Uncalled4 may have higher initialization or overhead costs that become proportionally less significant with larger workloads, while its core algorithm shows competitive efficiency at scale.

The largest gains were observed against Remora, where Fishnet achieved 20-22-fold speedups at 100 reads and 31-33-fold at 100000 reads, indicating that Fishnet scales more efficiently than Remora. A major reason for the extensive speedup is likely the multi-threaded implementation of the underlying processing compared to Remora, which operates in single-threaded mode. This architectural difference becomes increasingly pronounced as dataset sizes grow.

Overall, these results demonstrate that Fishnet is highly competitive across all tested scenarios and is particularly well suited for high-throughput applications and resource-constrained environments where GPU acceleration may not be available.

Alignment Fidelity

To evaluate alignment consistency, we compared normalized mean differences (NMDs) for 100000 alignments across all tool pairs. The largest differences were observed in the Fishnet-f5c and Remora-f5c comparisons, which showed nearly identical distributions (both median NMD of ~ 0.01%, maximum NMD of ~ 39%). The Fishnet-Uncalled4 and Remora-Uncalled4 comparisons exhibited similar mean differences (~ 0.01%) but with less pronounced outliers (maximum NMD ~ 2%). In contrast, Uncalled4 and f5c were more closely aligned with one another (median NMD ~ 0.001%). These patterns likely reflect algorithmic variation among tools, including differing parametrization of the dynamic programming routines and different preprocessing steps (Kovaka et al. 2025; Gamaarachchi et al. 2020). Despite the variability in magnitude, most alignments across all tools remained within 0.03% of the signal length, indicating an overall robustness of the tested signal-to-sequence alignment methods.

Against this background, the similarity between Fishnet and Remora stood out with a median NMD below 0.00007% and a maximum NMD of 0.13%. The reference-to-signal alignments showed a similarly tight correspondence between the two tools (median NMD of 0.00007%). This near perfect concordance is expected given that Fishnet was designed as a reimplementation of Remora's alignment algorithm. The minor discrepancies that do exist likely stem from floating-point inaccuracies across programming languages, which can accumulate throughout the iterative refinement steps. The fact that these differences remain extremely small even at the tail of the distribution strongly supports the fidelity of Fishnet's reimplementation of the Remora algorithm and its intended goal as a direct stand-in for Remora signal-to-sequence alignments.

Case Study: m^{1}A Signal Variation

Having established both speed and fidelity, we next assessed whether Fishnet could support biologically meaningful analyses. We applied it to analyze signal behavior around a known m^1A site in three different RNA sequence contexts between modified and unmodified sequences. Aligned signals revealed context-dependent differences between modified and unmodified reads, with the strongest effects in signal intensity (Cohen's d<-0.8 at five bases 5' of m^1A in *COX1* context). Notable differences in the dwell times were limited to individual sites (Cohen's d<-1.2 at two bases 5' of m^1A in *COX1* and Cohen's d<-1 at one base 5' of m^1A in *COX3*). Dimensionality reduction analyses revealed substantial overlap between modified and unmodified reads, with only small clusters where subsets of modified reads separate clearly.

These findings are consistent with prior analyses of other modifications, such as Pseudouridine or Inosine, reinforcing that modification signatures are highly context-dependent and often subtle (Makhamreh et al. 2024; Chen et al. 2023). This highlights the necessity for specialized and well-calibrated detection models, especially when considering more complex in-vivo data (Wu et al. 2024; Alagna et al. 2025; Yu et al. 2024; Teng et al. 2024).

Demonstrating Fishnet's Two-Command Workflow

The m^1A example successfully illustrates a practical use-case for Fishnet's intended workflow: signal-to-sequence alignment followed by an extraction of analysis-ready representations. These steps are performed with Fishnet's two subcommands: First, the *align* command performs complete signal-to-sequence mapping using only a BAM file, POD5 file(s), and a kmer-levels table. Produced alignments are written to PARQUET format for compact, fast and language-agnostic access. The interface enables users to easily switch between alignment types and DNA or direct RNA modes, accommodating for diverse sequencing data. Although the default settings aim to be broadly applicable, various parameters allow users to fine-tune the alignment process. Afterwards, the *reformat* command translates produced alignments into base-wise summary statistics (mean, median, standard deviation of the signal intensity, dwell time) or uniformly shaped signal vectors for modeling tasks. Integrated filtering by base position, motif or genomic region enables targeted analyses and minimizes the computational load.

This two-step process minimizes preprocessing overhead, reduces the need for custom scripts and ensures reproducible transformations, thus providing a streamlined and accessible analysis workflow.

Positioning Fishnet within the existing Ecosystem

The tools Fishnet were compared to occupy distinct niches in the broad ecosystem of signal-centric Nanopore data analysis. Remora focusses on generating training datasets for modification detection models, with no dedicated signal-to-sequencing command available. While its API exposes core signal alignment features, users must implement extensive pre- and postprocessing logic themselves. F5c extents Nanopolish with GPU acceleration, but requires multiple preparation steps (format conversions, indexing) and separates query-to-signal and reference-to-signal into different commands with distinct output formats. These design choices, along with additionally implemented commands, reflect f5c's role as an end-to-end methylation analysis framework rather than a lightweight alignment tool. Uncalled4 provides a broad environment for visualization, conversions, pore-model training and statistics. Its alignments can be generated in a single command but require explicit specification of kit and flowcell information, metadata that may be missing in large consortium or archival datasets and often requires additional tools to retrieve metadata from the POD5, either through an API or a graphical viewer (Dietrich et al. 2024).

In contrast, Fishnet emphasizes straightforward signal-to-sequence alignments with minimal setup, a lightweight interface and consistent outputs. No format conversions, index files, or detailed metadata are required. Instead of implementing full analysis or visualization toolchains, Fishnet focusses on fast and general-purpose alignments intended to serve as a foundation for downstream workflows.

Conclusions

Fishnet provides a concise and efficient alignment-centric workflow that complements existing tools. It offers a combination of high performance, near-perfect fidelity to Remora, and a streamlined two-command design. These properties lower entry barriers and facilitate reproducible and flexible high-throughput signal-to-sequence alignment. As such, Fishnet fits well with large-scale Nanopore signal-level investigations for modification detection purposes and beyond.

Methods

Splint ligation of RNA constructs

RNA constructs corresponding to the human mitochondrial genes *COX1*, *COX2*, and *COX3* were generated by splint ligation as previously described (Alagna et al. 2025). Briefly, RNA oligonucleotides were 5'-phosphorylated using T4 polynucleotide kinase (New England Biolabs, M0201) and purified using Oligo Clean & Concentrator Kit (Zymo Research, D4060). Ligation was performed using T4 RNA ligase 2 (New England Biolabs, M0239) with equimolar amounts of in vitro-transcribed RNA and synthetic oligonucleotides, together with a complementary DNA splint oligonucleotide (98% of RNA amount). After denaturation at 75°C and cooling to 25°C, the reaction was incubated at 16°C overnight. Following DNase I digestion and purification, constructs were polyadenylated using E. coli poly(A) polymerase (New England Biolabs, M0276). For detailed protocol, see (Alagna et al. 2025).

Benchmark data

Benchmarking both the processing speed and resulting alignments was performed on DNA sequencing data from the genome in a bottle dataset provided by ONT[3]. A subset of the provided POD5 files were downloaded and basecalled and mapped using Dorado basecaller (v0.9.0 + 9dc15a8). The data was mapped against the GRCh38 human reference provided by Gencode. From the raw POD5 and the basecalled and mapped BAM data, subsets containing short (≥ 100 & <7615 bases), medium (≥ 7615 &

<22306 bases) and long (≥ 22306 bases) reads were extracted. The boundaries were set so that each subset contained roughly the same number of reads. Each was subset again into groups containing 100, 1000, 10000 and 100000 reads, resulting in 12 benchmark datasets.

Comparing processing times

The command line benchmarking tool hyperfine (v1.19.0) was used to properly time the processing speed for a given benchmark run, each of which consisted of one warm-up run and five timed runs. Fishnet, Uncalled4 and f5c were benchmarked using default settings. The settings used for running Remora were adjusted to match the default settings of Fishnet (refinement iterations = 2, refinement algorithm = dwell-penalty, half-bandwidth = 5, rough rescale algorithm = theil_sen).

Both query- and reference-to-signal alignments were calculated for the twelve benchmark datasets (see Benchmark data). Remora was benchmarked single-threaded only, whereas the other tools were additionally timed with 8, 16 and 24 parallel threads. Overall, 312 benchmarking runs were performed.

For consistent system resources throughout different runs, the benchmarking was performed in slurm (v21.08.5) tasks with 32 CPUs and 64GB of memory (Yoo et al. 2003). The test system did not have a GPU, and as such f5c could only be tested using the CPU mode. This represents a notable limitation for the f5c comparison, as GPU acceleration is a prominent feature of this tool and CPU-only performance does not fully reflect its intended use case. The f5c results should therefore be interpreted primarily as a CPU-mode baseline rather that a full evaluation of the tool’s capabilities.

Comparing alignments

The systematic comparison of produced alignments was performed with the 100000 medium length read benchmark dataset. Query-to-signal alignments were calculated with Fishnet, Remora, f5c and Uncalled4. For the reference-to-signal alignment, only Fishnet and Remora were compared, since the produced *eventalign* tables that get generated by f5c and Uncalled4 only contain signal information for each base, but not the signal indices directly, and as such cannot be aligned to the original signal.

Where possible, the alignments were produced, and the generated output files were parsed into a uniform format. The separate files were merged into one dataset, where each row contains the alignments from all tools for a given read.

To quantify the degree of similarity between two alignments a and b, the normalized mean difference (NMD) was calculated. The NMD is defined as

$$NMD = \frac{\frac{1}{N}\sum_{i=0}^{N} |a_i - b_i|}{\max\left(a_{N-1}, b_{N-1}\right) - \min(a_0, b_0)}$$

Equation *1: Normalized mean difference (NMD) between two signal-to-sequence alignments a and b* where N corresponds to the number of aligned boundaries (number of bases + 1) and the denominator represents the total signal span. Normalization ensures comparability between reads of different lengths.

Analysis of m¹A sites in RNA constructs

For the oligos representing m^1A motifs in *COX1*, *COX2* and *COX3*, the reference sequence of the modified construct contains 141 bases, with a single m¹A at the 71st base. The unmodified construct consists of 155 bases, where the 78th base corresponds to the m¹A site in the modified construct. These center sites are flanked by 10 bases that are unique to each sample construct, and in turn, these are flanked by 32 bases that are identical for all sequences. Beyond this range, the sequence differs between the modified and unmodified variants.

The analysis was performed in the same way for all three samples. First, the reference-to-signal alignment was calculated using Fishnet's *align* command.

Signal visualization was prepared in a custom python script, where the alignment and signal were loaded, the signal standardized and subset to only contain bases of interest. Bases of interest included those located 52 bases up- and downstream from the central m¹A/A site (in each direction: 10 sample-specific bases + 32 constant bases + 10 bases that differ between modified and unmodified as a buffer). To filter extreme outliers, reads that contained measurements that deviate more than five standard deviations from the mean were skipped. The data was collected for 10000 reads to improve the readability of the generated plots and limit the amount of memory required while generating.

To analyze features derived from the alignments, the alignments were passed to Fishnet's *reformat* module with the *stats* approach, calculating the mean and standard deviation of the signal intensity, as well as the number of measurements (dwell time) for each base. Here 52 bases up- and downstream from the m¹A/A site were regarded. The calculated features were statistically compared between the modified and unmodified samples, using a two-sample Kolmogorov-Smirnov test at each base, with p-values adjusted using Bonferroni correction. Given the large sample sizes and resulting extremely small p-values (largest p-values $< 10^{-98}$ (*COX1*), $< 10^{-24}$ (*COX2*) and $< 10^{-19}$ (*COX3*); see Supplementary Table 2), Cohen's d was calculated for each base as a more interpretable, sample size-independent metric of the magnitude of differences between signals from modified and unmodified oligos (Cohen 2013).

Finally, to perform dimensionality reduction on the aligned signal, the signal around the m¹A/A site was interpolated into a uniform shape using Fishnet's *reformat* module with the *interpolate* approach, interpolating the signal chunk assigned to each base of interest into 30 samples. The interpolation was performed regarding only the central m¹A site, and including one, two and four bases up- and downstream from it. The different regions of interest were applied to inspect how well dimensionality reduction can separate the data between modified and unmodified with a varying amount of information. For each of these ranges of interest a separate *reformat* run was performed, adjusting the *positions-of-*

interest flag accordingly. Afterwards the modified and unmodified data was concatenated after subsetting the larger of the two data to the size of the smaller one. The combined and balanced data was then used to calculate a Uniform Manifold Approximation and Projection (UMAP) with two dimensions using the umap-learn Python package (v 0.5.9).

Alignment algorithm

The signal to sequence alignment algorithm implemented in Fishnet is adapted from the one used in ONT's Remora tool. Here the alignment process consists of two major steps: an initial alignment and an iterative refinement.

The initial alignment is constructed using the move table generated by the basecaller. This is an array of Boolean values that indicates when the sequencer detected a new base in the signal, represented by a 1. By combining this information with the sampling stride, which is stored with the move table, an alignment from positions in the basecalled (query) sequence to chunks of the raw signal is created.

If the read is mapped to a reference, the associated CIGAR string can be used to derive a reference-to-signal alignment. This is done by first computing a reference-to-query mapping based on the CIGAR operations and then translating it to signal coordinates via the query-to-signal mapping, followed by linear interpolation to obtain a dense signal alignment for each reference position.

The process solely utilizes the information from the move table, while not considering the actual signal. As the expected current intensities are known for each k-mer (k = the number of bases inside the pore at a given time), the signal to sequence alignment can be refined by comparing the expected intensity with the measured ones for each base and adjusting the alignment boundaries in a way that minimizes the deviation between the two.

The expected intensities are provided by ONT in k-mer level tables for their relevant chemistries. The tables contain expected measurements for each possible k-mer in standard units, meaning approximately a mean of 0 and a standard deviation of 1.

To make the signal comparable it needs to be standardized in the same way, which is done in an initial standardization step using a scale ($scale_0$) and shift ($shift_0$) parameter stored in the POD5 entry for a given read:

$$signal_{norm,i} = \frac{signal - shift_{i-1}}{scale_{i-1}}$$

Equation *2: Standardization of the signal in refinement iteration i*

With the signal comparable to the expected values the refinement itself can be started. This process consists of a banded dynamic programming approach that can be performed repeatedly to converge to an optimal alignment. Each iteration starts with constructing a constrained search space (band) that limits the bases that are considered for a given signal measurement to a set number up- and downstream from the currently assigned one (by default ± 5). Then the dynamic programming algorithm traverses the alignment space within the band, scoring each signal measurement against the expected intensity for a given base using the squared distance. In addition to the scores being calculated, a traceback is set up that allows for the reconstruction of an optimized alignment where the distance between the measured and expected intensities is minimized.

After each intermediate boundary optimization step, the standardization parameters are re-calculated based on the new boundaries using regression analysis. Here either Least Squares or Theil-Sen regression are implemented, the latter of which is the default as it is more robust against outliers (Sen 1968). The resulting $shift$ and $scale$ are then used in the next refinement iteration to standardize the signal based on the latest alignment. This way, the alignment is optimized repeatedly, converging to an optimal alignment with each iteration.

Optionally, a rough re-calibration step can be performed before the refinement process starts. Here new $shift$ and $scale$ parameters are calculated using percentiles of the measured and expected levels instead of the entire signal. This provides a computationally more efficient approach that brings measured and expected signal levels closer together, reducing the number of refinement iterations needed.

In the last iteration re-calibration of the standardization parameters is no longer performed, as the dynamic programming algorithm generates the final alignment, which gets returned.

Rust libraries

Fishnet uses several third-party Rust libraries. BAM file loading is handled by the Noodles crate (v0.99.0), a Rust-native bioinformatics input/output library. Reading and writing parquet files is handled by the arrow2 crate (v0.18.0). Writing to JSON format is handled by the serde_json crate (v1.0.141).

Initially, POD5 data handling was done using the experimental pod5-rs crate (v0.1.0)[4], which provides basic access to contained signals and metadata. But since this crate is still early in development, key features for efficient large-scale data handling are not implemented, including lazy loading and efficient (parallel) random access. As such, the current version of Fishnet uses a custom POD5 reader API that utilizes the arrow2 crate to read contained data. This implementation enables chunk-wise lazy loading, random access from multiple threads in parallel and efficient read-wise iteration through one or more files. Only the logic for the signal decompression is adapted from the pod5-rs crate.

Other crates for minor functions include thiserror (v2.0.11) for proper error handling, log4rs (v1.3.0) for logging, clap (v4.5.47), console (v0.16.0) and indicatif (0.18.0) for the command line interface, and crossbeam (v0.8.4) for parallelization.

Declarations

Ethics approval and consent to participate

Not applicable

Consent for publication

Not applicable

Funding

This work was partly funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation, project no. 439669440 TRR319 RMaP TP A07 (to S.G., L.L., and K.F.) and C04 (to S.P. and S.G.). S.G. and L.L. acknowledge funding from the Boehringer Ingelheim Stiftung.

Author Contribution

V.D. and L.L. conceived the idea. V.D. took the lead in writing the manuscript. L.L. sequenced the oligos. V.D., L.L. and S.P. interpreted the results and contributed to the manuscript. S.M. performed the splint ligation experiments. L.W. and K.F. designed and ordered the analyzed oligos. S.G. supervised the work and edited the manuscript. M.H. contributed to the planning and supervision of the work. All authors read and approved the final manuscript.

Acknowledgements

Not applicable

Data Availability

The Genome in a Bottle DNA data is openly available from ONT (https://epi2me.nanoporetech.com/giab-2025.01/). The direct RNA data is available on ENA with the Accession number PRJEB103800.The source code for Fishnet along with extensive documentation, and executable binaries are provided in the supplementary data (fishnet_main_repository.zip & fishnet_executables.zip). A separate repository contains all processing scripts that were used for data acquisition, benchmarking processing times, comparing alignments and analyzing the m1A contexts (fishnet_processing_repository.zip).

References

1. Alagna Nicolò, Mündnich S, Miedema J, et al. ModiDeC: A Multi-RNA Modification Classifier for Direct Nanopore Sequencing. Nucleic Acids Res. 2025;53(14):gkaf673. https://doi.org/10.1093/nar/gkaf673.

2. Boccaletto P, Stefaniak F, Ray A, et al. MODOMICS: A Database of RNA Modification Pathways. 2021 Update. Nucleic Acids Res. 2022;50(D1):D231–35. https://doi.org/10.1093/nar/gkab1083.
3. Chen H-X, Liu Z-D, Xin, Bai, et al. Accurate Cross-Species 5mC Detection for Oxford Nanopore Sequencing in Plants with DeepPlant. Nat Commun. 2025;16(1):3227. https://doi.org/10.1038/s41467-025-58576-x.
4. Chen L, Ou L, Jing X, et al. DeepEdit: Single-Molecule Detection and Phasing of A-to-I RNA Editing Events Using Nanopore Direct RNA Sequencing. Genome Biol. 2023;24(1):75. https://doi.org/10.1186/s13059-023-02921-0.
5. Chen X, Xu H, Shu X, Chun-Xiao S. Mapping Epigenetic Modifications by Sequencing Technologies. Cell Death Differentiation. 2025;32(1):56–65. https://doi.org/10.1038/s41418-023-01213-1.
6. Cohen J. A Power Primer. Psychol Bull. 1992;112(1):155–59. https://doi.org/10.1037/0033-2909.112.1.155.
7. Cohen J. 2013. *Statistical Power Analysis for the Behavioral Sciences*. 0 ed. Routledge. https://doi.org/10.4324/9780203771587
8. Cruciani S. and Eva Maria Novoa. 2025. The New Era of Single-Molecule RNA Modification Detection through Nanopore Base-Calling Models. *Nature Reviews Molecular Cell Biology*, ahead of print, October 13. https://doi.org/10.1038/s41580-025-00896-3
9. Diensthuber G, and Eva Maria Novoa. Charting the Epitranscriptomic Landscape across RNA Biotypes Using Native RNA Nanopore Sequencing. Mol Cell. 2025;85(2):276–89. https://doi.org/10.1016/j.molcel.2024.12.014.
10. Dietrich V, Alagna Nicolò, Helm M, Gerber S, and Tamer Butto. Pod5Viewer: A GUI for Inspecting Raw Nanopore Sequencing Data. Bioinformatics. 2024;40(12):btae665. https://doi.org/10.1093/bioinformatics/btae665.
11. Furlan M, Delgado-Tejedor A, Mulroney L, Pelizzola M, Novoa EM, and Tommaso Leonardi. Computational Methods for RNA Modification Detection from Nanopore Direct RNA Sequencing Data. RNA Biol. 2021;18(sup1):31–40. https://doi.org/10.1080/15476286.2021.1978215.
12. Gamaarachchi H, Lam CW, Jayatilaka G, et al. GPU Accelerated Adaptive Banded Event Alignment for Rapid Comparative Nanopore Signal Analysis. BMC Bioinformatics. 2020;21(1):343. https://doi.org/10.1186/s12859-020-03697-x.
13. Hewel C, Wierczeiko A, Miedema J, et al. Direct RNA Sequencing Enables Improved Transcriptome Assessment and Tracking of RNA Modifications for Medical Applications. Nucleic Acids Res. 2025;53(22):gkaf1314. https://doi.org/10.1093/nar/gkaf1314.
14. Jörg M, Plehn JE, Kristen M, et al. N1-Methylation of Adenosine (m1A) in ND5 mRNA Leads to Complex I Dysfunction in Alzheimer’s Disease. Molecular Psychiatry ahead print. January 2024;29. https://doi.org/10.1038/s41380-024-02421-y.
15. Kovaka S, Hook PW, Jenike KM, et al. Uncalled4 Improves Nanopore DNA and RNA Modification Detection via Fast and Accurate Signal Alignment. Nat Methods. 2025;22(4):681–91. https://doi.org/10.1038/s41592-025-02631-4.

16. Lee Y, Choe J, Park OH, and Yoon Ki Kim. Molecular Mechanisms Driving mRNA Degradation by m6A Modification. Trends Genet. 2020;36(3):177–88. https://doi.org/10.1016/j.tig.2019.12.007.

17. Lesbirel S, Wilson SA. 2019. The m6A–methylase Complex and mRNA Export. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms* 1862 (3): 319–28. https://doi.org/10.1016/j.bbagrm.2018.09.008

18. Li Q, Wang CSD, and Jizhong Lou. RMNet: An RNA m6A Cross-Species Methylation Detection Method for Nanopore Sequencing. Curr Drug Targets. 2025;26(11):799–812. https://doi.org/10.2174/0113894501405283250627072052.

19. Liu Y, Rosikiewicz W, Pan Z, et al. DNA Methylation-Calling Tools for Oxford Nanopore Sequencing: A Survey and Human Epigenome-Wide Evaluation. Genome Biol. 2021;22(1):295. https://doi.org/10.1186/s13059-021-02510-z.

20. Makhamreh A, Tavakoli S, Fallahi A, et al. Nanopore Signal Deviations from Pseudouridine Modifications in RNA Are Sequence-Specific: Quantification Requires Dedicated Synthetic Controls. Sci Rep. 2024;14(1):22457. https://doi.org/10.1038/s41598-024-72994-9.

21. Mao Y, Dong L, Liu X-M, et al. m6A in mRNA Coding Regions Promotes Translation via the RNA Helicase-Containing YTHDC2. Nat Commun. 2019;10(1):5332. https://doi.org/10.1038/s41467-019-13317-9.

22. Mayr JA, Tobias B, Haack P Freisinger, et al. Spectrum of Combined Respiratory Chain Defects. J Inherit Metab Dis. 2015;38(4):629–40. https://doi.org/10.1007/s10545-015-9831-y.

23. Pagès-Gallego M, Van Soest DMK, Nicolle JM, Besselink, et al. Direct Detection of 8-Oxo-dG Using Nanopore Sequencing. Nat Commun. 2025;16(1):5236. https://doi.org/10.1038/s41467-025-60391-3.

24. Richter U, Evans ME, Clark WC, et al. RNA Modification Landscape of the Human Mitochondrial tRNALys Regulates Protein Synthesis. Nat Commun. 2018;9(1):3966. https://doi.org/10.1038/s41467-018-06471-z.

25. Rübsam FNM, Liu-Wei W et al. Yu Sun,. 2025. MoDorado: Enhanced Detection of tRNA Modifications in Nanopore Sequencing by off-Label Use of Modification Callers. Preprint, February 22. https://doi.org/10.1101/2025.02.18.638820

26. Sen PK. Estimates of the Regression Coefficient Based on Kendall’s Tau. J Am Stat Assoc. 1968;63(324):1379–89. https://doi.org/10.1080/01621459.1968.10480934.

27. Shafik AM, Huiqing Z, Lim J, Dickinson B, Jin P. Dysregulated Mitochondrial and Cytosolic tRNA m1A Methylation in Alzheimer’s Disease. Hum Mol Genet. 2022;31(10):1673–80. https://doi.org/10.1093/hmg/ddab357.

28. Smits P, Smeitink J, Van Den L, Heuvel. Mitochondrial Translation and Beyond: Processes Implicated in Combined Oxidative Phosphorylation Deficiencies. J Biomed Biotechnol. 2010;2010:1–24. https://doi.org/10.1155/2010/737385.

29. Spangenberg J, Mündnich S, Busch A, et al. The RMaP Challenge of Predicting RNA Modifications by Nanopore Sequencing. Commun Chem. 2025;8(1):115. https://doi.org/10.1038/s42004-025-01507-

0.

30. Teng H, Stoiber M. Ziv Bar-Joseph, and Carl Kingsford. 2024. Detecting m6A RNA Modification from Nanopore Sequencing Using a Semi-Supervised Learning Framework. Preprint, January 7. https://doi.org/10.1101/2024.01.06.574484
31. Vujaklija I, Biđin Siniša, Marin Volarić, et al. Detecting a Wide Range of Epitranscriptomic Modifications Using a Nanopore-Sequencing-Based Computational Approach with 1D Score-Clustering. Nucleic Acids Res. 2025;53(1):gkae1168. https://doi.org/10.1093/nar/gkae1168.
32. White LK, Hesselberth JR. Modification Mapping by Nanopore Sequencing. Front Genet. 2022;13:1037134. https://doi.org/10.3389/fgene.2022.1037134.
33. Wu Y, Shao W, Liu S, et al. Simultaneous Profiling of ac4C and m5C Modifications from Nanopore Direct RNA Sequencing. Int J Biol Macromol. 2025;305(May):140863. https://doi.org/10.1016/j.ijbiomac.2025.140863.
34. Wu Y, Shao W, Mengxiao Yan, et al. Transfer Learning Enables Identification of Multiple Types of RNA Modifications Using Nanopore Direct RNA Sequencing. Nat Commun. 2024;15(1):4049. https://doi.org/10.1038/s41467-024-48437-4.
35. Xiong W, Zhao Y, Wei Z, et al. N1-Methyladenosine Formation, Gene Regulation, Biological Functions, and Clinical Relevance. Mol Ther. 2023;31(2):308–30. https://doi.org/10.1016/j.ymthe.2022.10.015.
36. Xu L, and Masahide Seki. Recent Advances in the Detection of Base Modifications Using the Nanopore Sequencer. J Hum Genet. 2020;65(1):25–33. https://doi.org/10.1038/s10038-019-0679-0.
37. Yoo AB, Morris A, Jette, Grondona M. 2003. SLURM: Simple Linux Utility for Resource Management. In *Job Scheduling Strategies for Parallel Processing*, edited by Gerhard Goos, Juris Hartmanis, and Jan Van Leeuwen, vol. 2862, edited by Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn. Lecture Notes in Computer Science. Springer Berlin Heidelberg. https://doi.org/10.1007/10968987_3
38. Yu B, Nagae G, Midorikawa Y, et al. m6ATM: A Deep Learning Framework for Demystifying the m6A Epitranscriptome with Nanopore Long-Read RNA-Seq Data. Brief Bioinform. 2024;25(6):bbae529. https://doi.org/10.1093/bib/bbae529.
39. Yu J, Chen M, Huang H, et al. Dynamic m6A Modification Regulates Local Translation of mRNA in Axons. Nucleic Acids Res. 2018;46(3):1412–23. https://doi.org/10.1093/nar/gkx1182.
40. Zhang C, and Guifang Jia. Reversible RNA Modification *N* 1-Methyladenosine (m1A) in mRNA and tRNA. Genomics Proteomics Bioinformatics. 2018;16(3):155–61. https://doi.org/10.1016/j.gpb.2018.03.003.
41. Zhang Y, Lu L, Li X. Detection Technologies for RNA Modifications. Experimental Molecular Medicine. 2022;54(10):1601–16. https://doi.org/10.1038/s12276-022-00821-0.

Footnotes

1. https://github.com/nanoporetech/dorado

2. https://github.com/nanoporetech/remora
3. https://epi2me.nanoporetech.com/giab-2025.01/
4. https://github.com/bsaintjo/pod5-rs

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- supplementarytable1processingtimes.xlsx
- supplementarytable2alignmentcomparison.xlsx
- fishnetexcutables.zip
- fishnetprocessingrepository.zip
- supplementaryfigures.pdf
- fishnetmainrepository2.zip