

## Point-by-point responses

We would like to thank the reviewers **R1** and **R2** for their thorough review of our manuscript. Please find our point-by-point response below.

**R1.General** This manuscript introduces DeepChopper, a genomic language model specifically designed to mitigate chimera artifacts in nanopore direct RNA sequencing. The authors leverage HyenaDNA, an efficient genome language model utilizing long-range implicit convolution, to generate output embeddings. This is coupled with a multi-layer perceptron (MLP) and a sliding window-based probability prediction mechanism, enhanced by majority voting for robust predictions. As the first genomic language model tailored for long-read sequencing data, this work marks a significant advancement in transcriptomics. It addresses a critical challenge in the field, showcasing its utility in improving data quality for applications such as transcript annotation and gene fusion detection. The study is innovative, methodologically sound, and a valuable contribution to genomic research. Below, I highlight the manuscript's strengths and suggest revisions to further improve its clarity and impact.

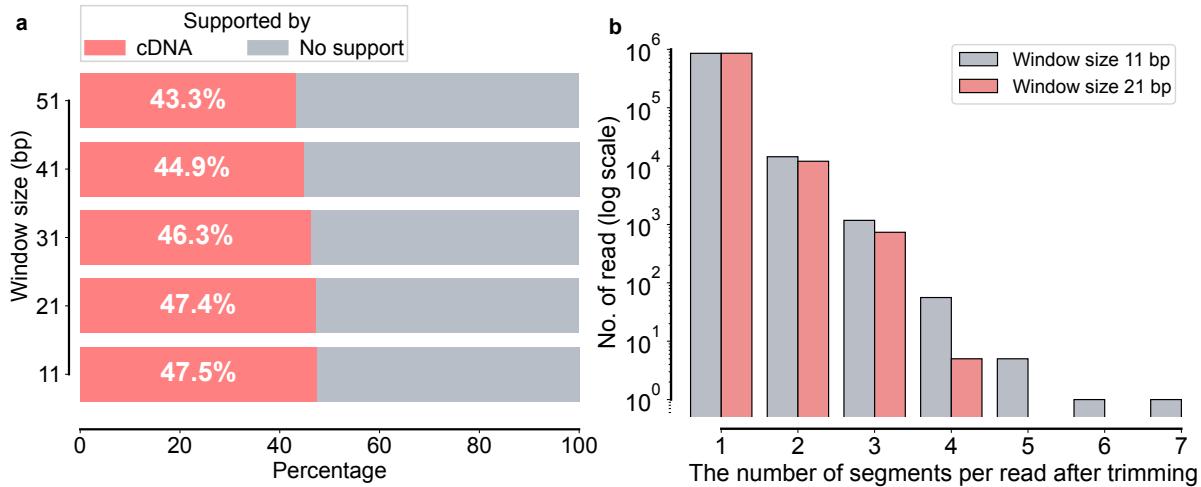
Strength:

1. DeepChopper is the first genomic language model tailored for long-read sequencing data. It provides an end to end pipeline for chimera affected DNA sequence task. Its ability to handle sequences up to 32K nucleotides, combined with its integration of sequencing quality scores, ensures precise detection of adapter sequences at single-nucleotide resolution.
2. The tool is rigorously validated across multiple platforms, including ONT and PacBio, and different chemistries (RNA002 and RNA004). Its robustness and practical impact on reducing false positives in gene fusion detection are well-documented.
3. The method is comparatively faster than traditional methods due to Hyena's long range efficient capabilities and efficient computational complexity.
4. Improvements in transcript annotation and artifact reduction have strong implications for advancing transcriptomics research.

**Response** We sincerely thank the reviewer for their thoughtful and constructive feedback on our manuscript introducing DeepChopper. We appreciate the recognition of our work's significance and innovation in addressing chimera artifacts in nanopore direct RNA sequencing. Below, we provide point-by-point responses to the reviewer's suggestions.

**R1.Q1** Sliding Window Rationale: While the 21-nucleotide sliding window approach achieves an effective balance between smoothing and sensitivity, further explanation of how this parameter was optimized would strengthen the methodology section. Clarifying whether alternative window sizes were tested and how their performance compared would provide additional insights.

**Response** In response to the reviewer's comment, we conducted additional tests using multiple window sizes to optimize our approach. These results are presented in Response Figure 1 and have been incorporated into the revised manuscript as a **new Extended Data Fig. 3**. We have also added detailed explanations of these tests in the revised manuscript (highlighted sections). As shown in Response Figure 1, we systematically evaluated window sizes of 11, 21, 31, 41, and 51 nucleotides to determine the optimal parameter. In Response Figure 1a, the red bars represent the



**Response Figure 1** (Manuscript Extended Data Fig. 4) **Effect of window size on chimeric alignment detection and read fragmentation.** (a) Analysis of different sliding window sizes (11, 21, 31, 41, and 51 nucleotides) showing the percentage of cDNA-supported chimeric alignments (red bars) in VCaP. Higher percentages indicate better support. (b) Distribution of the number of segments per read after trimming (x-axis) for window sizes 11 (gray) and 21 (pink), shown on a logarithmic scale (y-axis). Data represents sampling of 1M reads from VCaP dataset. Window size 21 maintains similar detection sensitivity to window size 11 while producing significantly fewer fragmented reads.

ratio of supporting chimeric alignments by direct cDNA data, with higher percentages indicating better support. Window sizes 11 and 21 showed the highest supporting ratios (47.5% and 47.4%, respectively), with larger windows showing progressively declining support (46.3%, 44.9%, and 43.3% for window sizes 31, 41, and 51). However, Response Figure 1b demonstrates a critical advantage of window size 21 over window size 11. While both detect similar percentages of chimeric alignments, window size 21 produces significantly fewer segments per read after trimming. This indicates that window size 21 maintains more contiguous predicted sequences compared to window size 11, which frequently fragments reads into 4-7 segments. This over-segmentation with window size 11 can potentially introduce artifacts and complicate downstream analysis. Window size 21 therefore represents the optimal balance point in our testing - maintaining the high detection sensitivity of smaller windows while preserving read continuity. Importantly, we recognize that optimal parameters may vary depending on specific dataset characteristics. For this reason, we have implemented a flexible approach by providing the parameter *-smooth-window* in our software, which allows users to adjust the window size according to their specific data requirements. Additionally, we have provided a detailed guide to change the *-smooth-window* and other command-line parameters in response to **R1.Q6**.

**R1.Q2** Other baseline models: Given that other Genome based language models exist (such as DNABERT, DNABERT2, The Nucleotide Transformers), authors might include results from these models as baselines or discussion of these models to better understand the capabilities of DeepChopper (in terms of accuracy).

**Response** We appreciate the reviewer's suggestion to include comparisons with existing genomic

language models. In response, we have added a comprehensive discussion in the revised manuscript (highlighted) outlining the key differences between DeepChopper and general-purpose genomic language models. As clarified, DeepChopper is specifically designed to address the unique challenges of long-read sequence analysis at single-nucleotide resolution. While models such as DNABERT [1], DNABERT2 [2], and Nucleotide Transformer [3] are valuable for a broad range of genomic tasks, their context length limitations—approximately 512 bp for DNABERT, 10,000 bp for DNABERT2, and 6,000 bp for Nucleotide Transformer—make them suboptimal for long-read applications. In contrast, DeepChopper processes sequences up to 32 kilobases, a context length specifically tailored to accommodate most full-length mRNA transcripts.

Given these fundamental architectural differences, direct performance comparisons are not appropriate. DeepChopper’s design further enables single-nucleotide resolution, a critical feature for base-level precision that general-purpose models are not optimized to deliver. The revised discussion now clearly delineates DeepChopper’s unique strengths in the context of long-read sequence analysis, while acknowledging the broader utility of existing genomic language models.

**R1.Q3** Handling of Long Reads: The manuscript notes that sequences exceeding 32K nucleotides are excluded from analysis. Additional details on the prevalence of such sequences in datasets and the implications of this constraint would help users understand the tool’s potential limitations.

**Response** Thanks for this important comment regarding the handling of ultra-long reads. We have revised the manuscript (highlighted) to provide a comprehensive explanation of the 32K context length limitation and its practical implications.

The 32K limit was introduced based on both architectural and biological considerations. Architecturally, it reflects the context window limitation of the underlying HyenaDNA model [4]. While extending this limit is technically possible, doing so would substantially increase computational demands without clear benefits for the vast majority of use cases.

Biologically, the manuscript now clarifies that human mRNA transcripts have a median length of 1.5–2kb with a peak around 2,065 bp [5], and even complex genes rarely exceed 10–15 kb. Notably, the longest known transcripts, such as titin (TTN), range from 20–25 kb—well within our 32K limit. We also cite nanopore RNA-seq data showing that the maximum aligned read lengths are typically 21 kb, with most reads falling well below this threshold[6].

These updates should provide readers with a clear understanding of the rationale and relevance of the 32K context limit for most long-read transcriptomic applications.

**R1.Q4** Performance with RNA004 Chemistry: The reduced efficacy of DeepChopper with RNA004 compared to RNA002 (21% vs. 91% reduction in chimeric alignments) is an important observation. Including a discussion on differences in adapter designs or sequencing chemistry and their potential impact on performance would add depth to the analysis.

**Response** We appreciate the reviewer highlighting this important difference in performance between RNA002 and RNA004 chemistries. In response, we have expanded our manuscript (highlighted) to address this important observation. Notably, DeepChopper was trained exclusively on RNA002 data, which was the predominant sequencing chemistry available during model development. Despite this training limitation, we deliberately included RNA004 evaluation to assess DeepChopper’s zero-shot transfer capability to newer chemistries without additional fine-tuning. The observed difference in chimeric alignment reduction (91% for RNA002 vs. 21% for RNA004) reflects both the model’s training bias toward RNA002 patterns and fundamental differences in the sequencing chemistries themselves.

RNA004 represents a significant advancement in Oxford Nanopore’s chemistry, specifically designed to produce cleaner reads with fewer artifacts, including chimeric sequences [7]. This inherent improvement in the RNA004 protocol means that raw RNA004 data already contains substantially fewer chimeric artifacts than RNA002 data, giving DeepChopper less opportunity for further improvement. In other words, the smaller relative improvement in RNA004 should be contextualized by the already superior quality of the input data. We have added this explanation to the manuscript (highlighted) to provide readers with a more comprehensive understanding of the performance differences and to highlight DeepChopper’s zero-shot generalization ability across different sequencing chemistries.

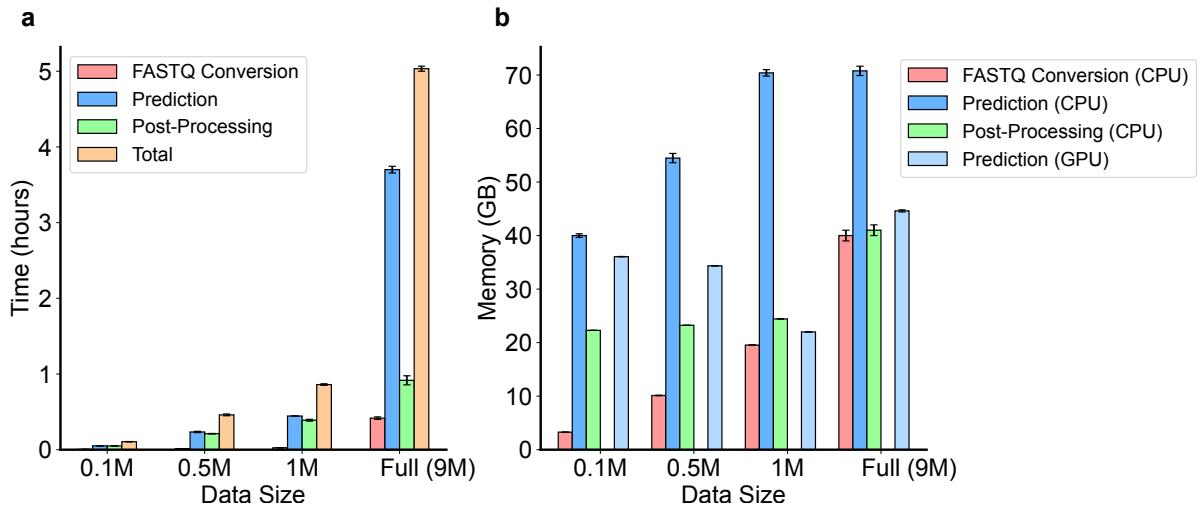
**R1.Q5** Computational Resource Documentation: Providing a table summarizing runtime, memory, and GPU/CPU requirements for datasets of varying sizes would be highly beneficial for researchers planning large-scale analyses.

**Response** We thank the reviewer for this valuable suggestion. In response, We have expanded the documentation of computational resource in the revised manuscript by including a **new Extended Data Fig. 4** (Response Figure 2) that illustrates the scaling of runtime and memory requirements across different dataset sizes. As shown in Response Figure 2, we benchmarked DeepChopper’s performance on four dataset sizes (0.1M, 0.5M, 1M, and the full 9M reads) to provide researchers with clear expectations for planning their analyses. The results show that DeepChopper can process a full 9-million-read dataset from the VCaP cell line in approximately **5 hours**.

We have also included a detailed breakdown of resource usage for each processing step:

1. **FASTQ Conversion:** This initial stage shows minimal resource requirements for smaller datasets but scales to 25 minutes and 40 GB CPU memory for the full 9M read dataset.
2. **Adapter Prediction:** This most computationally intensive stage benefits significantly from GPU acceleration, requiring 3 hours and 42 minutes on our full dataset with 33 GB GPU memory and 71 GB CPU memory.
3. **Post-Processing:** This stage scales linearly, requiring 55 minutes and 41 GB CPU memory for the full dataset.

The benchmarks were conducted using high-performance infrastructure (16 CPU cores, dual NVIDIA A100 GPUs), and we have included these specifications in the **Methods** section (highlighted) to provide complete transparency regarding the computing environment. These details have been incorporated into the new “**Computational Benchmarks**” subsection under Methods. These comprehensive benchmarks demonstrate that DeepChopper remains practical for large-scale sequencing projects, efficiently processing millions of reads within hours while maintaining reasonable memory requirements.



**Response Figure 2 (Manuscript Extended Data Fig. 3) Computational performance metrics across different data sizes from the VCaP cell line dRNA-seq.** (a) Runtime analysis showing processing time requirements for different pipeline stages (FASTQ Conversion, Prediction, Post-Processing) and total runtime across four dataset sizes : subsampled (0.1M, 0.5M, 1M) and full (9M) reads derived from the VCaP dRNA-seq. As data size increases, prediction time becomes the dominant component, with the full dataset requiring approximately 5 hours of total processing time. (b) Memory usage comparison between CPU and GPU implementations across the same data sizes. The prediction stage shows consistently higher memory requirements, with CPU memory usage for prediction reaching approximately 70 GB for the larger datasets. All measurements include error bars representing standard deviation from three runs.

**R1.Q6** Generalization Guidelines: Including recommendations for parameter tuning in diverse datasets or scenarios would enhance the tool’s applicability across different research contexts.

**Response** We thank the reviewer for this valuable suggestion. In response, We have created a dedicated “Parameter Optimization Guidelines” (<https://github.com/ylab-hi/DeepChopper/blob/main/documentation/parameters.md>) section in our GitHub documentation to clarify DeepChopper’s applicability across different research contexts. This documentation provides detailed recommendations for optimizing four key command-line parameters used in the post-processing step:

1. Sliding Window Size (*-smooth-window*): We recommend the default value of 21 nucleotides for most applications, as it optimally balances detection sensitivity and read continuity as shown in our comparative analysis (Response Figure 1).
2. Minimum Interval Size (*-min-interval-size*): This post-smoothing parameter (default: 13) can be adjusted based on adapter characteristics, with lower values increasing sensitivity to short adapter fragments.
3. Maximum Process Intervals (*-max-process-intervals*): This parameter (default: 4) limits adapter regions processed per read and can be reduced to decrease fragmentation or increased for complex libraries.

4. Minimum Read Length (*-min-read-length*): This final filtering parameter (default: 20) prevents the generation of extremely short fragments and can be adjusted based on the application's requirements.

These revisions directly address the reviewer's suggestions and contribute to improving the overall clarity and utility of the manuscript. We thank the reviewer again for their thoughtful feedback, which has helped refine and strengthen the work.

**R2.General** This manuscript presents DeepChopper, a deep-learning-based tool designed to identify and remove chimera artifacts in Nanopore direct RNA sequencing (dRNA-seq). The authors propose that sequencing artifacts, particularly internal adapters, introduce errors that affect downstream analyses such as transcript annotation and gene fusion detection. DeepChopper leverages HyenaDNA, a genomic language model originally trained on DNA sequences and integrates sequencing quality scores to improve adapter detection. The study evaluates DeepChopper using synthetic training data and real dRNA-seq datasets, claiming a significant reduction in chimeric reads compared to ONT's Dorado basecaller.

**Response** We appreciate the reviewer's thoughtful and constructive feedback on our manuscript presenting DeepChopper. Below, we provide detailed responses to each point raised.

**R2.Q1** The manuscript does not clearly explain why a DNA-trained genomic language model (HyenaDNA) is appropriate for RNA sequencing data. RNA has different sequence structures, error profiles, and biological contexts than DNA. Since HyenaDNA was trained specifically on human DNA, it likely requires fine-tuning to adapt to RNA sequencing artifacts. The authors should provide a stronger theoretical justification for why this model is expected to work and discuss whether any modifications were made to improve its performance on RNA data. In addition, if non-human dRNA-seq data were to be analysed, DeepChopper, although not evaluated, will likely to fail.

**Response** We thank the reviewer for raising this important point regarding the use of a DNA-trained genomic language model (HyenaDNA [4]) for RNA sequencing data. We provide the following clarifications:

- **Adapter Sequences:** The adapter sequences that we detect are synthetic DNA sequences ligated to RNA molecules during library preparation (as discussed in Ref.3 [8] of the manuscript). Since these adapters are DNA-based, a DNA-trained model is appropriate for this task.
- **Model Transferability:** HyenaDNA is designed to learn nucleotide-level genomic patterns, which are transferable to RNA sequences due to the shared nucleotide composition between DNA and RNA. Genomic language models capture contextual relationships between nucleotides, making them effective for both DNA and RNA analysis (as detailed in Ref.8 [9] of the original manuscript).
- **Fine-Tuning on RNA Data:** To ensure optimal performance, we extensively fine-tuned HyenaDNA using RNA-specific data, allowing it to adapt to the unique error profiles and structural characteristics of Nanopore dRNA-seq. This fine-tuning process enables the model to effectively learn RNA-specific patterns while preserving its core nucleotide understanding (see “**Training Data Preparation**” in the Methods section).

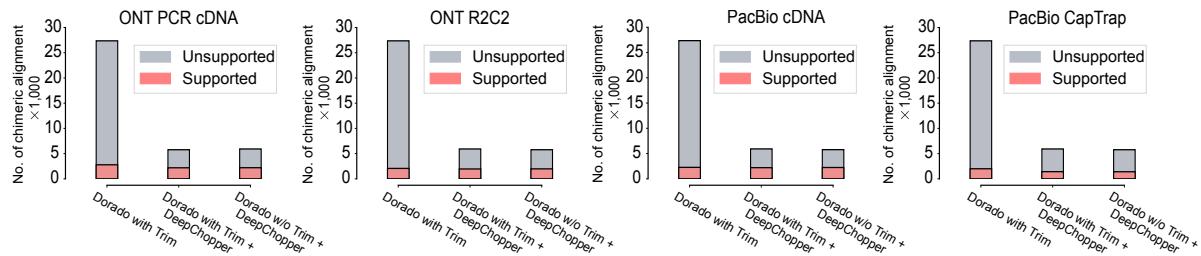
Furthermore, **R2** expressed concern that “*if non-human dRNA-seq data were to be analyzed, DeepChopper will likely fail.*”

We thank the reviewer for raising this important point. While the reviewer expressed concern that DeepChopper may not generalize to non-human dRNA-seq data, our original manuscript already included an analysis demonstrating its applicability beyond human datasets.

As presented in Extended Data Fig. 2 of the original submission (now updated and relabeled as **Extended Data Fig. 5** in the revised manuscript), we applied DeepChopper to dRNA-seq data from the **F121-9 mouse embryonic stem cell line** provided by the LRGASP consortium. In this analysis, DeepChopper effectively removed chimeric read artifacts that lacked support from cDNA-based long reads across multiple sequencing platforms, confirming its utility in non-human species.

To further strengthen this point and address the reviewer’s concern related to **R2.Q7**, we expanded the figure in the revised manuscript to include new benchmarking results using **Dorado adapter trimming followed by DeepChopper**, as shown in **Extended Data Fig. 5** and Response Figure 3 below. These results demonstrate that DeepChopper consistently removes unsupported chimeric alignments in both Dorado trimmed and untrimmed read sets.

Together, these results support the generalizability of DeepChopper to non-human dRNA-seq data without requiring species-specific retraining.



**Response Figure 3** (Manuscript Extended Data Fig. 5) Chimeric alignments from direct RNA sequencing (dRNA-seq) of the F121-9 cell line (mouse), evaluated for support using additional Oxford Nanopore Technologies (ONT) and Pacific Biosciences (PacBio) sequencing data with different protocols. DeepChopper-involved methods reduce unsupported chimeric alignments across all methods compared to Dorado with adapter trimming. The bar colors indicate chimeric alignments supported by additional sequencing data (red) and those lacking support (grey).

**R2.Q2** DeepChopper is introduced as a novel method, but the manuscript does not compare it to existing adapter trimming and chimera detection tools such as Porechop, Pychopper, or ONT’s Dorado with adapter trimming enabled (rDNA-seq data was base-called with “–no-trim” option). Without direct comparisons in terms of accuracy, precision, recall, F1-score, and runtime, it is unclear whether DeepChopper offers meaningful advantages. A benchmarking analysis is necessary to determine whether DeepChopper outperforms or complements these existing methods.

**Response** We appreciate the reviewer’s comments regarding the need for benchmarking and comparison of DeepChopper against existing adapter trimming and chimera detection tools. We would like to respectfully clarify that DeepChopper addresses challenges specific to Oxford Nanopore Technologies (ONT) **direct RNA sequencing (dRNA-seq)**—a protocol that is fundamentally distinct from those targeted by the tools mentioned.

## 1. Protocol-Specific Relevance

**DeepChopper** is specifically designed to identify and trim adapter sequences in ONT **direct RNA sequencing** data. In contrast, tools such as **Pychopper** [10] are optimized for **cDNA-based protocols** and use models trained on cDNA adapter sequences, which are not present in dRNA-seq datasets. Therefore, a direct comparison is not appropriate, as Pychopper is not built to process or accurately identify dRNA-seq adapters.

## 2. Limitations of Legacy Tools

**Porechop**, another tool cited, is no longer actively maintained (last update over eight years ago) and was developed prior to the introduction of ONT's dRNA-seq technology [11]. Our analysis demonstrates that Porechop is unable to detect adapters in dRNA-seq reads, making a direct comparison with DeepChopper infeasible (see Response Figure 4).

## 3. Benchmarking on Synthetic dRNA-seq Data

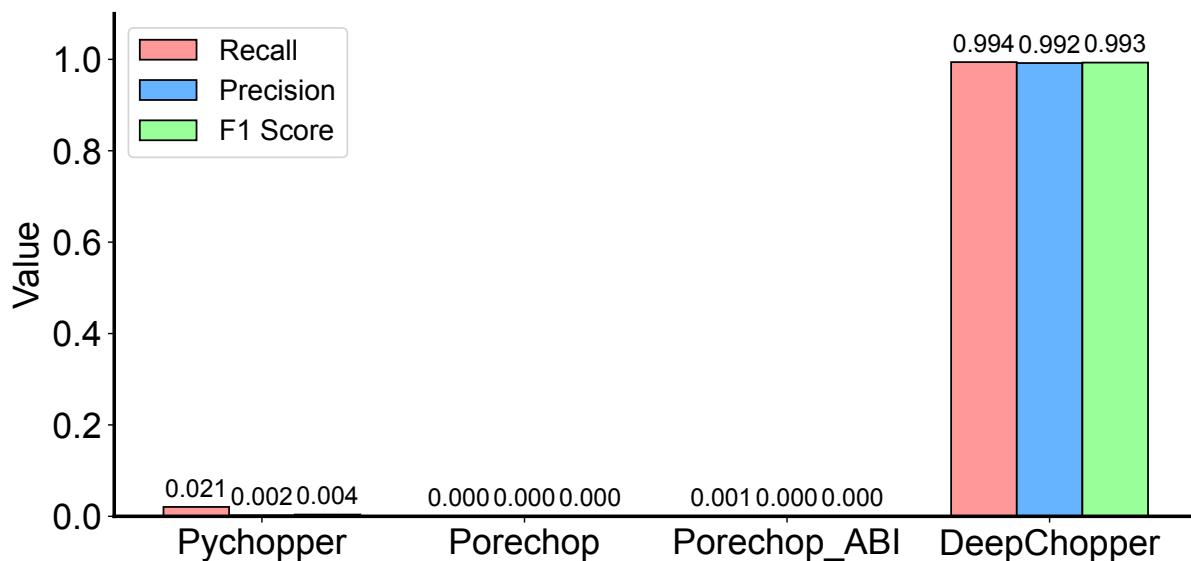
To address the concern, we benchmarked DeepChopper's performance against a range of adapter trimming tools, including those explicitly mentioned by the reviewer (**Pychopper**, **Porechop**) as well as an additional, more recent extension tool, **Porechop\_ABI** [12]. While Porechop\_ABI represents an advancement over the original Porechop, none of these tools are specifically designed for the characteristics of ONT **direct RNA sequencing** (dRNA-seq) data.

We performed benchmarking using a **synthetic dRNA-seq dataset** generated in the original manuscript ( $N = 60,000$  reads), which includes known adapter sequences introduced at defined positions. This design allows for robust performance evaluation in terms of precision, recall, and F1-score. As shown in Response Figure 4, **all existing tools**, including **Porechop\_ABI**, failed to accurately detect dRNA-seq adapter sequences. Their performance metrics were negligible, indicating a fundamental incompatibility with the dRNA-seq protocol.

In contrast, **DeepChopper accurately identified and trimmed both terminal and internal adapters**, highlighting its unique capability to handle the complexities of direct RNA reads. These findings reinforce that current tools are insufficient for adapter detection in dRNA-seq, and underscore the need for a purpose-built solution such as DeepChopper.

## 4. Additional Benchmarking with Dorado

We have also included a detailed comparison with **Dorado (with adapter trimming enabled)** in our response to **R2.Q7**, where we show that DeepChopper offers superior sensitivity and specificity for detecting dRNA-seq adapters, particularly in complex cases involving chimeric reads.



**Response Figure 4** (Manuscript Extended Data Fig. 1) Performance evaluation in a held-out test ( $N = 60,000$ ) dataset showing Recall, Precision, and F1 values for DeepChopper, Pychopper, Porechop, and Porechop\_ABI.

We have updated the manuscript to include these benchmarking results as **Extended Data Fig. 1** (Response Figure 4) and added **Extended Data Table 1** (Response Table 1), which provides a comparative summary of the capabilities and limitations of current tools versus DeepChopper. These results support the conclusion that existing tools are not well-suited for adapter trimming in dRNA-seq, underscoring the need for a dedicated solution.

**Response Table 1** (Manuscript Extended Data Table 1) Summary of Adapter Trimming Tools for analyzing dRNA-seq data

Adapter trimming tool	dRNA-seq terminal adapter trimming	dRNA-seq internal adapter trimming	Trimming existing dRNA-seq datasets (post-basecalling)
Porechop [11]	✗	✗	✗
Porechop_ABI [12]	✗	✗	✗
Pychopper [10]	✗	✗	✗
Dorado [13]	✓	✗	✗
DeepChopper (This study)	✓	✓	✓

Note: ✓ indicates the tool supports this functionality; ✗ indicates the tool does not support this functionality.

We hope these clarifications and new data address the reviewer's concerns regarding the novelty and benchmarking of DeepChopper.

**R2.Q3** The training data were created by manually inserting adapters into reads and identifying adapter regions using minimap2 soft-clipping. This process assumes that all 3' soft-clipped regions

correspond to adapter sequences, which may not be true. Furthermore, chimeric reads were artificially generated, which does not necessarily reflect real ONT sequencing artifacts. The authors should validate their training data approach by analyzing real dRNA-seq datasets to determine whether internal adapters appear at similar frequencies and distributions. In addition, the authors used minimap2 (a long-read aligner) to generate true training data. Why don't use alignment-based method directly? How the genomic language model will be better than alignment-based methods? Basically, all the benchmarks (e.g. counting chimeric reads, etc.) were all based on alignment results. This needs to be justified.

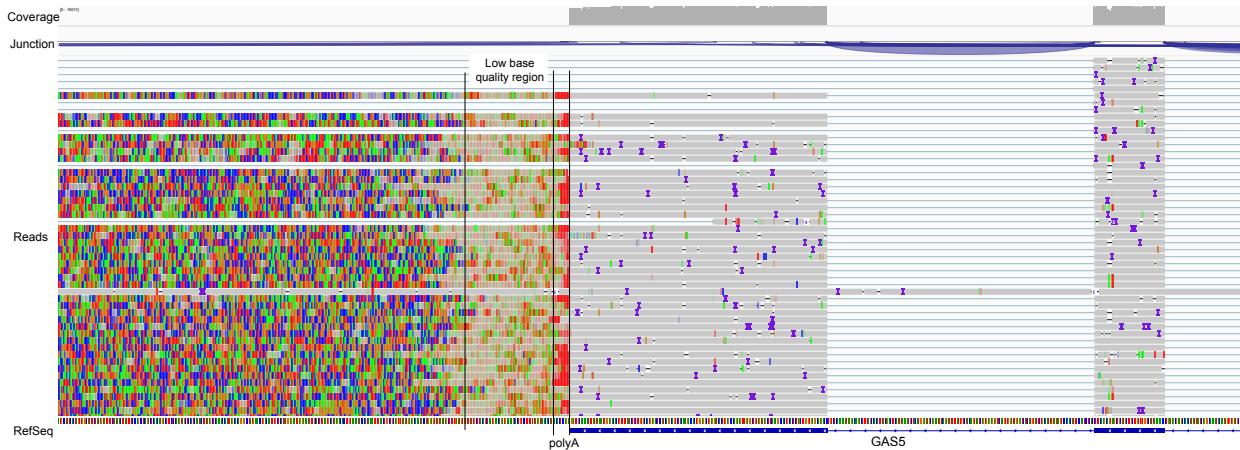
**Response** We thank the reviewer for the detailed comments and the opportunity to clarify our training data generation methodology and the rationale behind our modeling approach.

### 1. Refinement in Training Data Generation Using Biological Context

The reviewer notes that our training data were created by manually inserting adapters and identifying adapter regions using minimap2 soft-clipping, which may not fully capture the complexity of real dRNA-seq artifacts. We appreciate this concern and would like to clarify that **we did not assume all 3' soft-clipped regions to be adapter sequences**. Instead, our method incorporates a biologically informed refinement:

We specifically anchored on polyA tails—a reliable indicator of transcript termination in dRNA-seq—to guide the identification of downstream adapter sequences. Only soft-clipped regions of low base-quality that followed a polyA stretch were considered candidate adapter segments. This strategy substantially reduces the risk of falsely labeling endogenous soft-clipped sequences as adapter-derived, ensuring higher-quality training labels.

To support this strategy, we provide an example from real dRNA-seq data (Response Figure 5). In this example, a read corresponding to the GAS5 gene does not terminate at the expected polyA region (highlighted in red) but continues with a sequence segment of notably lower base quality (labeled “Low base quality region”). These low-quality regions extend into the soft-clipped part of the alignment, strongly indicating the presence of adapter-containing chimera artifacts. The figure clearly demonstrates how these artifacts display a consistent sequence pattern—well-aligned regions followed by polyA stretches, which are then followed by poorly aligned, low-quality segments (shown in the gray shaded area in Response Figure 5) that correspond to the adapter sequences and subsequent incorrectly captured fragments. This pattern recognition approach significantly reduces the risk of misclassifying non-adapter regions as adapters and ensures our training data more accurately reflects the biological context of direct RNA sequencing.



**Response Figure 5 Visualization of adapter-containing chimeric reads in dRNA-seq of the VCaP cell line.** This alignment visualization shows characteristic features of chimeric artifacts in dRNA-seq data. Individual reads (horizontal lines) display a consistent pattern: well-aligned sequences (colored bases) followed by internal polyA stretches (labeled “polyA”), which are immediately followed by regions of significantly lower base quality (labeled “Low base quality region”). These low-quality regions, extending into the soft-clipped portions (gray shaded area), represent adapter sequences and incorrectly captured fragments. This distinctive pattern—internal polyA followed by low-quality regions—provides a reliable signature for discriminating technical artifacts from genuine biological chimeras.

## 2. Justification for Using a Genomic Language Model Over Alignment-Based Methods

The reviewer also questions why alignment-based methods (e.g., minimap2) were not used directly for detecting chimeric artifacts, especially since alignment results were employed in benchmarking.

We clarify that while **alignment is useful for initial labeling and benchmarking, it is not sufficient for accurate artifact detection**. This is because:

- Not all chimeric alignments are artifacts.** Some represent biologically meaningful events, such as trans-splicing or gene fusion events.
- Alignment-based tools lack the context to distinguish artifact from biology.** Without modeling sequence composition, adapter motifs, or base quality patterns, traditional aligners cannot differentiate true gene fusions from chimera artifacts.
- Our genomic language model captures the latent patterns and statistical properties** of genuine transcripts versus synthetic chimeric sequences, including nucleotide distribution, quality scores, and adapter motifs.
- This modeling approach enables DeepChopper to selectively filter artificial chimeras while preserving biologically relevant reads,** which is crucial in accurate transcript annotation and minimizing false-positive gene fusion calls (see Fig. 3 in the manuscript).

Thus, while alignment was used in training and validation, our approach offers a learned, sequence-aware alternative that is capable of generalizing beyond rule-based alignment heuristics.

We have revised the manuscript to provide further details on these points and clarify the polyA-anchored strategy in the **Methods: Training Data Preparation** section.

**R2.Q4** DeepChopper classifies bases as adapters if the probability score is greater than 0.5, but the manuscript does not justify this threshold. If probability scores are clustered around 0.5, the classification may be unreliable. The authors should provide a distribution of probability scores and assess whether different cutoffs affect performance. An ROC curve or precision-recall analysis would help determine the optimal threshold for adapter detection.

**Response** We thank the reviewer for pointing out the ambiguity regarding the classification threshold used by DeepChopper. We agree that the description in the original manuscript—specifically the reference to a fixed threshold of 0.5—may have led to confusion.

To clarify, **DeepChopper does not use a fixed threshold of 0.5 for classification**. Instead, it employs a *softmax*-based multi-class classification approach, where the model outputs per-base probabilities for two classes: adapter and non-adapter. During inference, classification is performed by selecting the class with the higher probability at each nucleotide position, i.e., a base is assigned to the adapter class if  $P(y_i = \text{adapter}) > P(y_i = \text{non-adapter})$ , and vice versa. This approach inherently avoids reliance on a fixed threshold and ensures consistent interpretation of probabilistic outputs, especially in the context of mutually exclusive class labels.

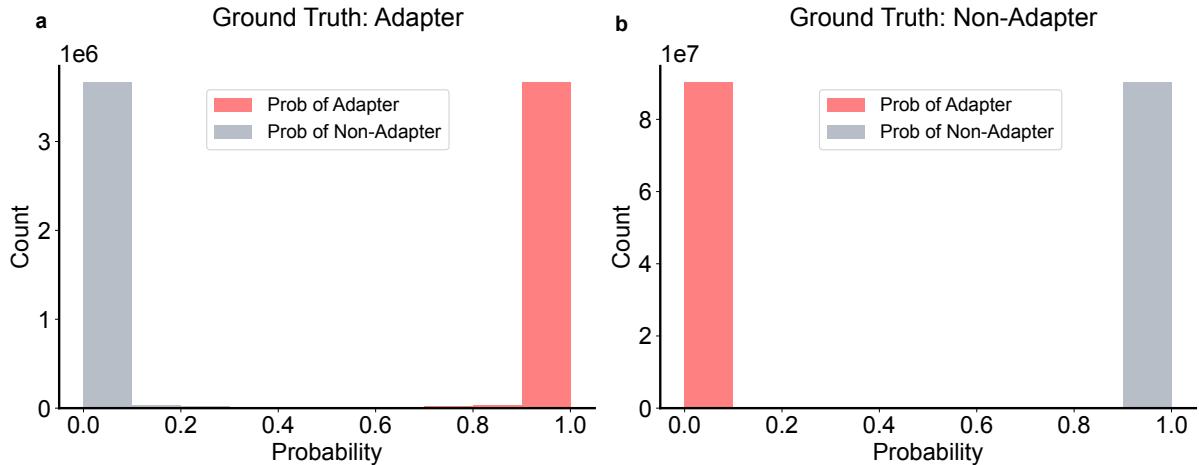
We have updated the manuscript to correct this discrepancy and provide a more accurate description of the classification head. Specifically, we have revised the relevant sentence in the *Language Model Architecture* subsection of the **Methods** to reflect the use of softmax rather than sigmoid, and to clarify the decision rule applied during inference.

In response to the reviewer's suggestion, we have also performed additional analyses to characterize the model's prediction confidence and assess the reliability of its classifications. We now include a new figure (Response Figure 6) that visualizes the distribution of predicted probabilities for the adapter and non-adapter classes on a held-out test set ( $N = 60,000$  reads). These analyses show that:

1. For true adapter bases (Response Figure 6a), the model assigns high adapter probabilities (mostly  $> 0.8$ ), and correspondingly low non-adapter probabilities.
2. For true non-adapter bases (Response Figure 6b), the model confidently predicts high non-adapter probabilities ( $> 0.8$ ), with low adapter probabilities ( $< 0.2$ ).
3. The overall probability distributions are strongly bimodal, indicating decisive predictions and minimal ambiguity between classes.

This sharp separation in probability space demonstrates robust feature learning and generalization.

We sincerely appreciate the reviewer's valuable feedback, which has helped us improve both the clarity and completeness of our manuscript.



**Response Figure 6** (Manuscript Extended Data Fig. 2) **Prediction probability distributions of DeepChopper for the held-out test dataset ( $N = 60,000$ )**. (a) Distribution of prediction probabilities for sequences with ground truth adapter classification. Red bars represent the probability of adapter prediction, while gray bars show the probability of non-adapter prediction. The count (y-axis) is shown in millions of sequences ( $10^6$  scale). (b) Distribution of prediction probabilities for sequences with ground truth non-adapter classification. Red bars indicate the probability of adapter prediction, while gray bars show the probability of non-adapter prediction. The count (y-axis) is shown in tens of millions of sequences ( $10^7$  scale). Both distributions demonstrate strong polarization toward correct classification probabilities, indicating the model’s high confidence in distinguishing between adapter and non-adapter sequences.

**R2.Q5** The manuscript introduces a Quality Block to incorporate sequencing base quality scores but it does not clearly explain how this improves model performance. Similarly, the sliding window refinement approach lacks detailed methodology, including how window size and majority voting criteria were selected. The authors should provide an ablation study testing the impact of the Quality Block and a sensitivity analysis for the sliding window parameters.

**Response** We appreciate the reviewer’s thoughtful comments and the opportunity to clarify the methodology and impact of both the Quality Block and the sliding window refinement procedure. We have revised the manuscript to include additional details and supporting analyses, which are now highlighted in the updated version.

**Quality Block Contribution** The Quality Block is designed to incorporate base-level sequencing quality scores into DeepChopper’s prediction pipeline, enhancing its robustness in noisy regions of the input sequence—particularly near ambiguous adapter boundaries, which exhibit low overall quality (See Response Figure 5).

In response to the reviewer’s suggestion, we performed an ablation study using an independent dataset ( $N = 100,000$  reads). As shown in Response Table 2, incorporating the Quality Block improved the F1 score from 0.97 to **0.99**, indicating a notable improvement in classification performance attributable to this component. This enhancement reflects the ability of quality signals to disambiguate difficult predictions, particularly in low-confidence regions. We have added the training setup to the revised Methods section and included the results of this ablation study as **Extended Data Table 2** in the revised manuscript.

**Response Table 2** (Manuscript Extended Data Table 2) Ablation Study Results for Quality Block

Model Configuration	F1 Score
With Quality Block	<b>0.99</b>
Without Quality Block	0.97

**Sliding Window Refinement** To further improve prediction smoothness and reduce isolated misclassifications, DeepChopper applies a post-hoc sliding window refinement step based on majority voting. We have now expanded the Methods section to detail both the rationale and parameter selection strategy.

We evaluated multiple window sizes (ranging from 11 to 51 nucleotides) on VCaP, balancing detection sensitivity with read continuity (See **R1.Q1**). As shown in Response Figure 1a, window sizes of 11 and 21 performed similarly in terms of raw prediction support (47.5% and 47.4%, respectively). However, as illustrated in Response Figure 1b, a window size of **21** significantly reduced the number of read fragments introduced by trimming, thereby preserving better overall sequence integrity while maintaining sensitivity. Based on this analysis, window size 21 was selected as the default.

To ensure flexibility for different datasets and sequencing characteristics, we have implemented the *-smooth-window* parameter in the command-line interface, allowing users to customize the window size as needed. Detailed documentation for this feature is available in DeepChopper’s online user guide and in our response to Reviewer 1, Comment 6 (See **R1.Q6**).

**Majority Voting Mechanism** The sliding window refinement applies a straightforward majority voting strategy over a centered window of size  $W$ , defined as follows:

$$y_i = \begin{cases} 1 & \text{if } \sum_{j=i-k}^{i+k} p_j > \frac{W}{2} \\ 0 & \text{otherwise} \end{cases}$$

Here,  $y_i$  is the final refined prediction at nucleotide position  $i$ ,  $k = \frac{W-1}{2}$  is the window radius, and  $p_j$  represents the initial binary prediction at position at  $j$ , where 1 indicates adapter and 0 indicates non-adapter. This approach smooths the basewise predictions by consolidating local prediction context, reducing isolated false positives and negatives.

We thank the reviewer for prompting these improvements, which have enhanced the clarity and rigor of our methodology. The relevant sections in the manuscript have been updated and highlighted accordingly.

**R2.Q6** The manuscript states that DeepChopper has 4.6 million parameters, suggesting it is computationally efficient but it does not provide any runtime or memory benchmarks. There is no information on how long DeepChopper takes to process a dataset, what GPU/CPU resources are required, or how it scales with large sequencing datasets. A computational efficiency evaluation is necessary to determine whether DeepChopper is practical for large-scale sequencing projects.

**Response** We thank the reviewer for this important suggestion. To address the concern regarding computational efficiency, we have conducted a comprehensive benchmarking analysis and expanded the relevant documentation in the revised manuscript.

As shown in Response Figure 2, we evaluated DeepChopper’s runtime and memory usage across datasets ranging from 0.1 million to 9 million reads. For the full 9 million-read VCaP dataset, DeepChopper completes processing in approximately **5 hours**. We also provide a detailed breakdown of resource requirements for each stage of the pipeline:

1. **FASTQ Conversion:** 25 minutes, peak **40 GB** CPU memory
2. **Adapter Prediction:** 3 hours and 42 minutes, peak **33 GB** GPU memory and **71 GB** CPU memory
3. **Post-processing and Sliding Window Refinement:** 55 minutes, peak **41 GB** CPU memory

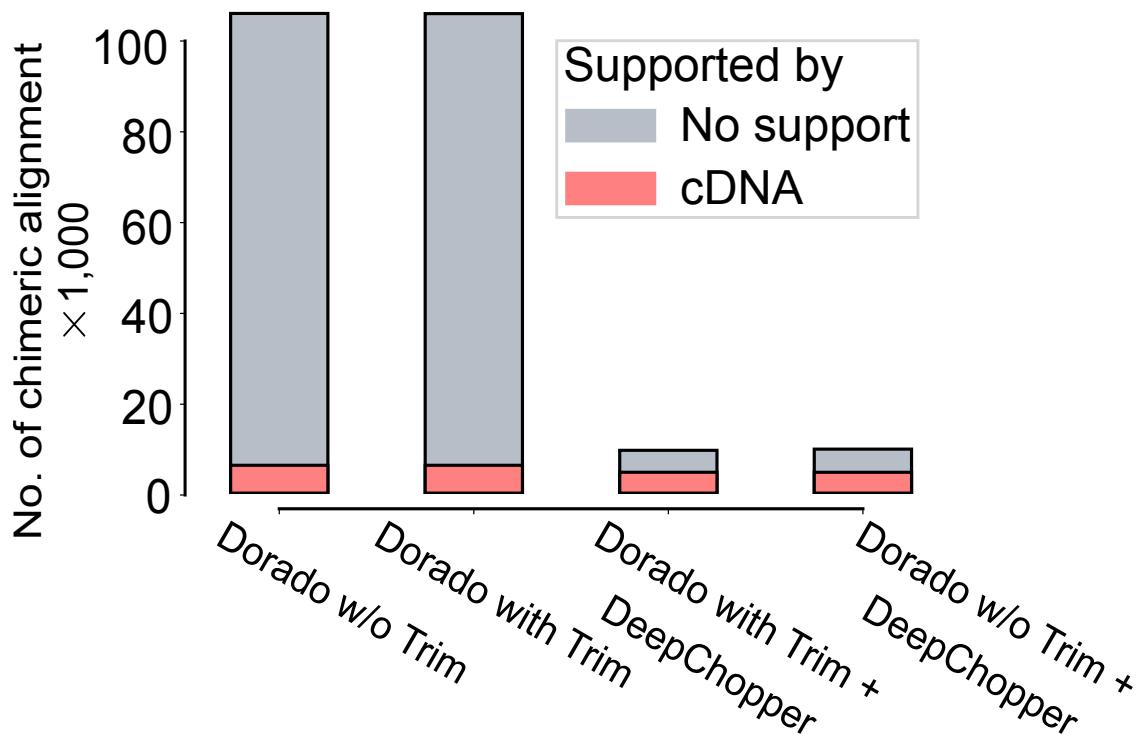
These benchmarks were performed on a system with **16 CPU cores** and **two NVIDIA A100 GPUs**. The hardware specifications and software environment have been added to the Methods section (highlighted) for full transparency.

This analysis demonstrates that DeepChopper scales efficiently to large datasets, with runtime and memory usage remaining practical for large-scale sequencing applications. The full benchmarking results, including scalability trends, are also discussed in our response to Reviewer 1, Comment 5 (See **R1.Q5**).

**R2.Q7** The validation dRNA-seq data were processed using Dorado with the “`--no-trim`” option, which preserves adapters and likely increases the observed chimeric read count. This choice could artificially inflate the problem DeepChopper aims to solve, making the model appear more effective than it actually is. The authors should repeat their analysis using Dorado with adapter trimming enabled to assess whether DeepChopper still provides a significant advantage.

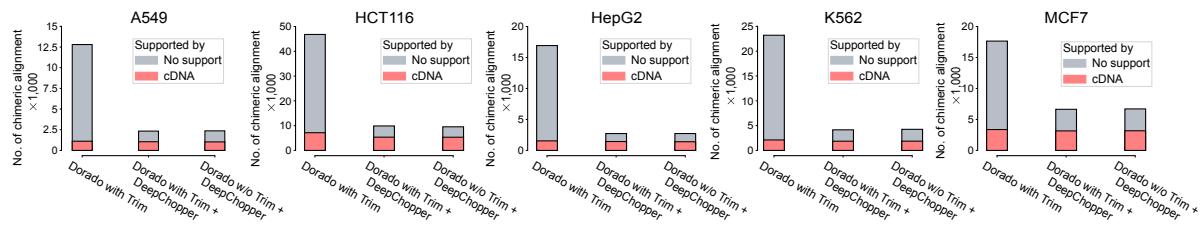
**Response** We thank the reviewer for raising this important concern. The potential for inflated chimeric read counts due to the use of Dorado with the `--no-trim` option is well taken, and we appreciate the opportunity to clarify and further substantiate DeepChopper’s effectiveness.

In response to the reviewer’s suggestion, we extended our benchmarking and downstream analyses by incorporating results from Dorado with adapter trimming enabled, both alone and in combination with DeepChopper. In our analysis of VCaP dRNA-seq data, we observed that Dorado alone—regardless of whether trimming was applied—was insufficient to eliminate spurious chimeric alignments. In contrast, DeepChopper substantially reduced unsupported chimeric events, whether applied independently or following Dorado trimming, as validated by orthogonal cDNA sequencing (Response Figure 7).

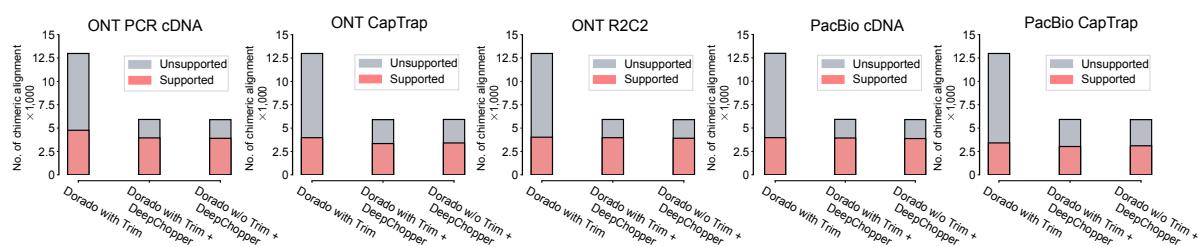


**Response Figure 7** (Manuscript Fig. 2d) Chimeric alignments (in thousands) for VCaP dRNA-seq reads processed by Dorado with and without adapter trimming, Dorado with adapter trimming followed by DeepChopper, and DeepChopper. DeepChopper-involved methods including Dorado with adapter trimming followed by DeepChopper and DeepChopper greatly reduce chimeric alignments not supported by direct cDNA sequencing.

To further generalize these findings, we performed an expanded analysis across six additional human cell lines (A549, HCT116, HepG2, K562, MCF7, and WTC11) as well as the F121-9 mouse embryonic stem cell line. As shown in Response Figure 8, Response Figure 9, and Response Figure 3, DeepChopper consistently reduced chimeric artifacts across all cases, including when applied after Dorado’s adapter trimming. These results reinforce that DeepChopper offers distinct and additive benefits beyond what is achieved through adapter trimming alone.

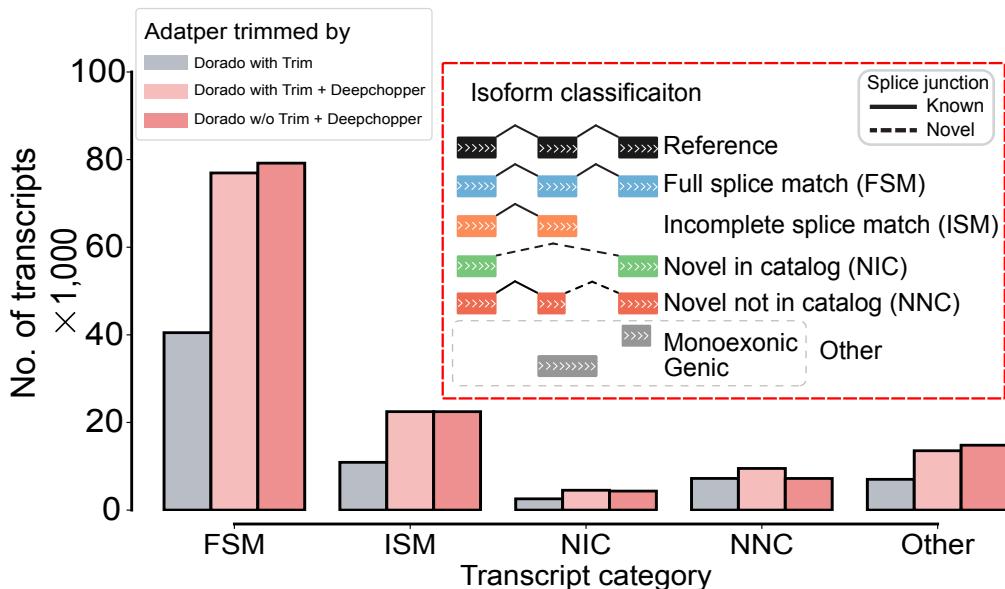


**Response Figure 8** (Manuscript Fig. 2g) The number of chimeric alignments (in thousands) for A549, HCT116, HepG2, K562, and MCF7 cell lines processed by Dorado with adapter trimming, Dorado with adapter trimming followed by DeepChopper and DeepChopper. DeepChopper-involved methods including Dorado with adapter trimming followed by DeepChopper and DeepChopper consistently reduce chimeric alignments not supported by cDNA sequencing across all cell lines.



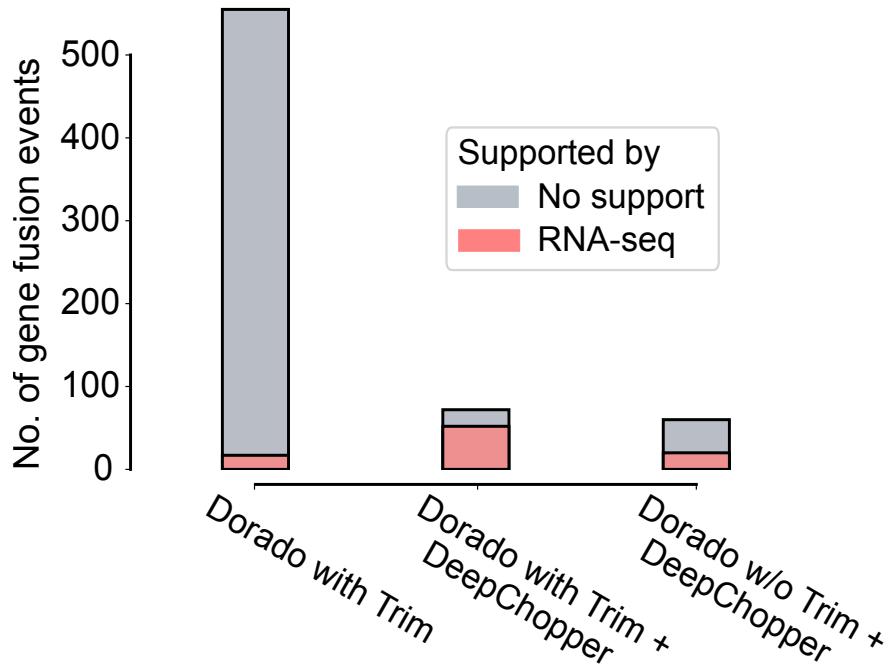
**Response Figure 9** (Manuscript Fig. 2h) Chimeric alignments from dRNA-seq of the WTC11 cell line, evaluated for support using additional ONT and PacBio sequencing data with different protocols. DeepChopper-involved methods including Dorado with adapter trimming followed by DeepChopper and DeepChopper reduce unsupported chimeric alignments across all methods compared to Dorado with adapter trimming.

We also evaluated the impact of DeepChopper on isoform-level transcript detection using VCaP cells. As illustrated in Response Figure 10, applying DeepChopper following Dorado trimming led to a greater number of detected transcripts across multiple isoform classification categories compared to Dorado trimming alone. This supports the complementary role of DeepChopper in enhancing downstream transcriptomic interpretation, even when applied after standard trimming procedures.



**Response Figure 10** (Manuscript Fig. 3b) The bar plot shows the number of transcripts (in thousands) across different isoform classification categories. DeepChopper-processed reads result in a higher number of transcripts compared to Dorado-trimmed reads. The inset details the isoform classification scheme.

Furthermore, we repeated our gene fusion analysis using Dorado-trimmed reads followed by DeepChopper. As shown in Response Figure 11, fusion detection was markedly improved in both DeepChopper-based pipelines—with or without prior trimming—demonstrating greater concordance with gene fusions identified from matched short-read RNA-seq data. This underscores DeepChopper’s ability to eliminate spurious fusion calls while preserving true, biologically meaningful fusion events.



**Response Figure 11** (Manuscript Fig. 3c) Detected gene fusions from Dorado adapter-trimmed reads and DeepChopper-processed reads using Dorado with adapter trimming followed by DeepChopper and DeepChopper. Gene fusions identified from short-read RNA-seq were used to validate fusion events detected from dRNA-seq.

Collectively, these findings confirm that DeepChopper's effectiveness is not contingent upon the inclusion of untrimmed adapter sequences. Its capacity to reduce chimeric artifacts, enhance isoform detection, and improve gene fusion calling remains robust across diverse datasets and preprocessing workflows. We have incorporated these updated analyses and figure references into the revised manuscript to directly address this concern.

## References

- [1] Yanrong Ji et al. “DNABERT: Pre-Trained Bidirectional Encoder Representations from Transformers Model for DNA-language in Genome”. In: *Bioinformatics* 37.15 (Aug. 9, 2021). Ed. by Janet Kelso, pp. 2112–2120. ISSN: 1367-4803, 1367-4811. DOI: [10.1093/bioinformatics/btab083](https://doi.org/10.1093/bioinformatics/btab083).
- [2] Zhihan Zhou et al. *DNABERT-2: Efficient Foundation Model and Benchmark for Multi-Species Genome*. 2023. DOI: [10.48550/arXiv.2306.15006](https://doi.org/10.48550/arXiv.2306.15006). arXiv: [2306.15006 \[cs, q-bio\]](https://arxiv.org/abs/2306.15006). Pre-published.
- [3] Hugo Dalla-Torre et al. “Nucleotide Transformer: building and evaluating robust foundation models for human genomics”. In: *Nature Methods* (2024), pp. 1–11.
- [4] Eric Nguyen et al. “Hyenadna: Long-range genomic sequence modeling at single nucleotide resolution”. In: *Advances in neural information processing systems* 36 (2024).
- [5] Inês Lopes et al. “Gene size matters: an analysis of gene length in the human genome”. In: *Frontiers in Genetics* 12 (2021), p. 559998.
- [6] Rachael E Workman et al. “Nanopore native RNA sequencing of a human poly (A) transcriptome”. In: *Nature methods* 16.12 (2019), pp. 1297–1305.
- [7] Charlotte Hewel et al. “Direct RNA sequencing enables improved transcriptome assessment and tracking of RNA modifications for medical applications”. In: *bioRxiv* (2024), pp. 2024–07.
- [8] Martin A Smith et al. “Molecular barcoding of native RNAs using nanopore sequencing and deep learning”. In: *Genome research* 30.9 (2020), pp. 1345–1353.
- [9] Gonzalo Benegas et al. *Genomic Language Models: Opportunities and Challenges*. 2024. arXiv: [2407.11435 \[q-bio.GN\]](https://arxiv.org/abs/2407.11435). URL: <https://arxiv.org/abs/2407.11435>.
- [10] *epi2me-labs/pychopper: cDNA read preprocessing*. Github. URL: <https://github.com/epi2me-labs/pychopper>.
- [11] Ryan R. Wick et al. “Completing bacterial genome assemblies with multiplex MinION sequencing”. In: *Microbial Genomics* 3.10 (Oct. 2017). ISSN: 2057-5858. DOI: [10.1099/mgen.0.000132](https://doi.org/10.1099/mgen.0.000132). URL: [http://dx.doi.org/10.1099/mgen.0.000132](https://dx.doi.org/10.1099/mgen.0.000132).
- [12] Quentin Bonenfant, Laurent Noé, and Hélène Touzet. “Porechop\_ABI: discovering unknown adapters in Oxford Nanopore Technology sequencing reads for downstream trimming”. In: *Bioinformatics Advances* 3.1 (2023), vbac085.
- [13] Oxford Nanopore PLC. *Dorado*. <https://github.com/nanoporetech/dorado>. 2023.