Resolving systematic bias in nanopore-based RNA modification detection

Alexandra Sneddon[1,2,3,4,*], Stefan Prodic[1,2,3], Eduardo Eyras[1,2,3,4]

1 EMBL Australia Partner Laboratory Network at the Australian National University, Canberra ACT 2601, Australia.
2 Centre for Computational Biomedical Sciences, The John Curtin School of Medical Research, Australian National University, Canberra ACT 2601, Australia.
3 The Shine-Dalgarno Centre for RNA Innovation, The John Curtin School of Medical Research, Australian National University, Canberra ACT 2601, Australia.
4 The Australian Research Council Centre of Excellence for the Mathematical Analysis of Cellular Systems, Australia.

*Correspondence to: Alexandra.Sneddon@anu.edu.au

Abstract

Nanopore direct RNA sequencing has accelerated the growth of the epitranscriptomics field, yet nanopore-based RNA modification landscapes are generally unreliable due to the heuristic and unvalidated quantification algorithm underpinning the standard, widely used method, modkit. We introduce a general benchmarking framework that uncovers systematic and often extreme bias in modkit's default implementation, which misleads biological conclusions, and we identify optimised settings that rescue modkit performance across multiple modification types and biological contexts, establishing an accurate and reliable standard for RNA modification quantification. A Nextflow workflow is available at https://github.com/comprna/modkitopt.

Main

Nanopore direct RNA sequencing enables single-molecule detection of native RNA modifications at single-nucleotide resolution[1]. Modification quantification pipelines typically employ Oxford Nanopore Technologies' (ONT) dorado[2] to call modified bases per read, and modkit[3] to aggregate dorado's per-read predictions into site-level stoichiometry, which is defined as the fraction of transcripts that are modified at a given reference position. In downstream analyses, stoichiometry cutoffs are applied to classify sites as modified or unmodified, thereby necessitating accurate stoichiometry estimation and thresholding for correct biological interpretation. However, modkit's parameters are estimated heuristically per dataset, without assessing their impact on site-level accuracy, and no quantitative approach exists to identify the optimal stoichiometry cutoff.

To estimate stoichiometry, modkit *pileup* first filters out dorado's low-confidence modification calls. By default, modkit estimates the confidence threshold by considering modification calls for a subset of reads, defining confidence as the maximum predicted probability (modified or non-modified) for each call, and then setting the threshold at the 10th percentile of these

confidence values across all calls. Since this heuristic is not based on prediction accuracy, threshold selection is vulnerable to being insufficiently stringent, resulting in elevated false discovery rates[4], or overly stringent, excluding true sites. Further, the default approach of identifying a single threshold value to filter both canonical and modified bases risks biasing filtering for rare modification types such as pseudouridine (pseU), where canonical base calls dominate and thus bias the threshold estimation. Although users can manually specify separate confidence thresholds for canonical and modified bases (*--filter-threshold* and *--mod-threshold,* respectively), identifying which values to use relies on subjective interpretation of the data or ad-hoc strategies, and no assessment of parameter performance is provided to guide a principled threshold selection.

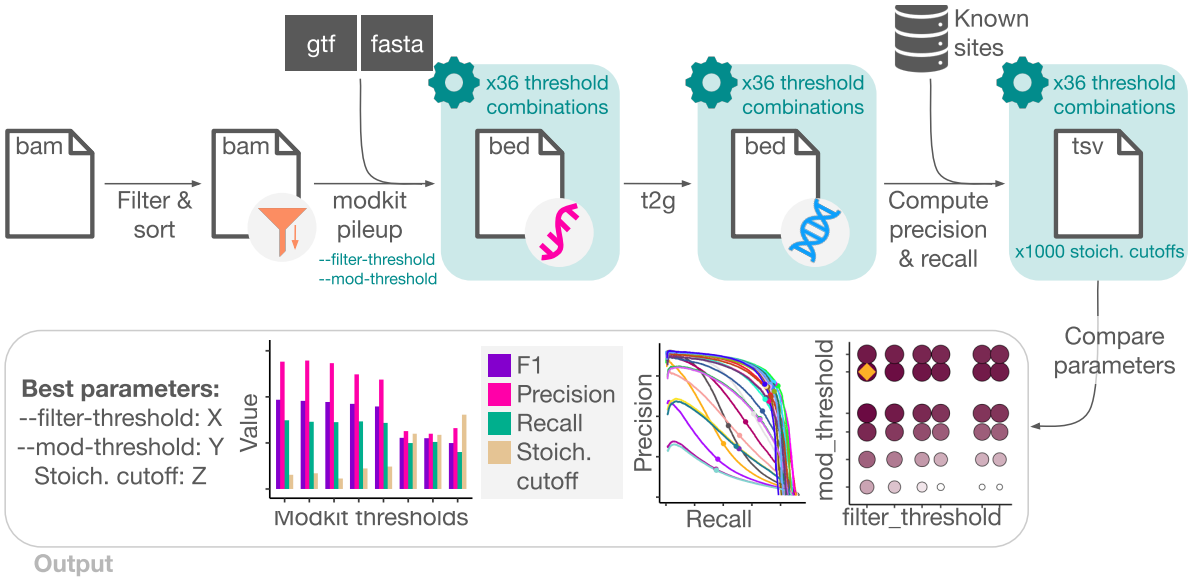

Figure 1. Overview of ModkitOpt. A transcriptome-aligned modBAM file containing per-read modification calls is filtered and sorted, and then processed with modkit *pileup* across a grid of filtering parameters (*--filter-threshold* and *--mod-threshold*) to generate site-level stoichiometry estimates. Transcript sites are converted to genomic coordinates and compared against validated modification sites across 1,000 stoichiometry cutoffs. Precision, recall and F1 score are visualised for each parameter combination, as well as for the default modkit implementation, and the thresholds and stoichiometry cutoff that maximise F1 score are identified and reported.

Here, we introduce ModkitOpt (**Figure 1**) (**Methods**), a general benchmarking framework that identifies the optimal modkit filtering parameters (*--filter-threshold* and *--mod-threshold*) and downstream stoichiometry cutoff for any given nanopore dataset. Taking as input a modBAM file produced by dorado, ModkitOpt exhaustively scans all combinations of 36 candidate modkit thresholds, and 1,000 candidate stoichiometry cutoffs, and quantifies the precision and recall of predicted sites against a set of validated sites, comparing performance across the parameter space to the modkit default.

ModkitOpt reports the optimal thresholds and stoichiometry cutoff to the user, and outputs plots that visualise modkit's performance in terms of precision (fraction of predicted sites that are in the validated set), recall (fraction of validated sites that are predicted) and F1 score ($2 \cdot \frac{(Pecision \cdot Recall)}{(Pecision + Recall)}$) across the parameter space, displaying for each threshold combination the

stoichiometry cutoff that maximised F1 score. Validated sites are provided for mammalian m6A and pseU, derived from GLORI-identified m6A sites in HEK293T cells[5] and BACS-identified pseU sites in HeLa cells[6], respectively. The user can also provide validated sites for other modifications, enabling ModkitOpt to be broadly applied. ModkitOpt is distributed as a Nextflow workflow, enabling reproducible execution on local machines or HPC systems, with resource allocation automatically managed for efficient and scalable analysis.

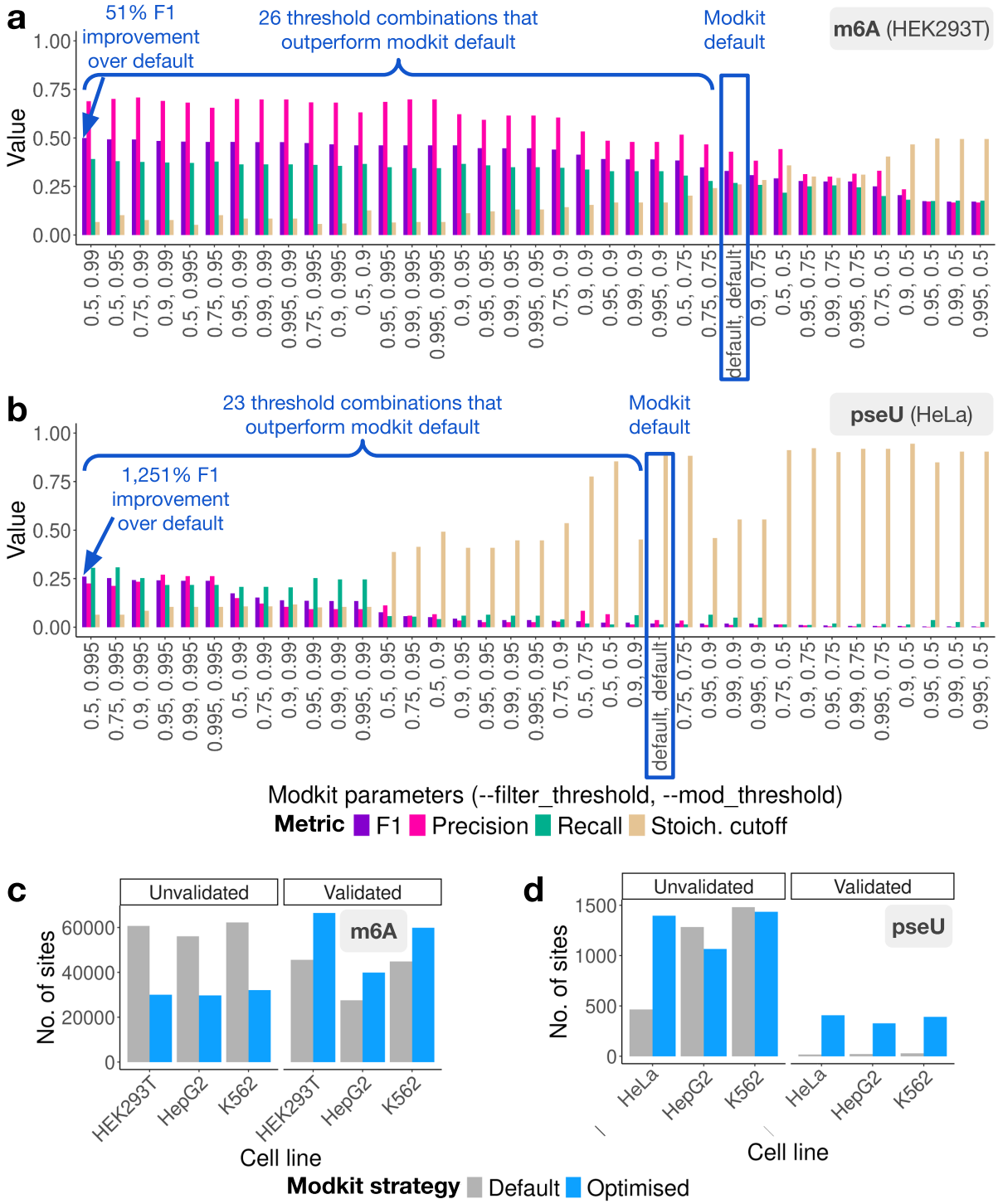

Figure 2. a,b ModkitOpt visualisation of modkit performance (y-axis) across 36 threshold combinations (x-axis), compared to the modkit default (annotated), in terms of F1 score (purple),

precision (pink) and recall (green), for N6-methyladenosine (m6A) in HEK293T cells (**a**) and for pseudouridine (pseU) in HeLa cells (**b**). Shown for each threshold combination is the performance for the stoichiometry cutoff (tan) that maximised F1 score. **c,d** Number of detected sites (total, unvalidated, validated) using default (grey) or optimised (blue) modkit thresholds, for m6A sites in HEK293T cells (**c**) and pseU sites in HeLa cells (**d**). The stoichiometry cutoff that maximised F1 score was used for each modkit strategy.

We first assessed whether modkit *pileup's* default heuristic algorithm accurately quantifies modified sites. For consistency, we ran ModkitOpt using modBAM files from the same cell line as the validated sites, HEK293T for m6A[5] and HeLa for pseU[6]. For both modification types, modkit's default was substantially less accurate than most of the tested thresholds, with the default implementation ranking 27th and 24th out of 37 parameter combinations for m6A (**Figure 2a**) and pseU (**Figure 2b**), respectively, in terms of F1 score. The optimal thresholds (m6A: *--filter-threshold=0.5, --mod-threshold=0.99*; pseU: *--filter-threshold=0.5, --mod-threshold=0.995*) significantly outperformed the default, achieving F1 score improvements of 51% for m6A and a remarkable 1,251% for pseU, relative to the default. The dramatic performance improvement for pseU reflects the inherent bias underlying modkit's heuristic threshold estimation, which disproportionately affects rarer modification types.

There were clear performance trends across the parameter space. The F1 score, which quantifies the precision-recall trade-off, was dominated by *--mod_threshold*, with more stringent values improving precision at the expense of recall. Performance for pseU was maximised at a more stringent *--mod_threshold* than for m6A, consistent with the lower confidence of dorado pseU calls that necessitates stricter filtering. For both modification types, the optimal stoichiometry cutoff increased across threshold combinations as F1 score decreased, suggesting that looser *--mod_threshold* values require stricter stoichiometry cutoffs to mitigate false positives. The impact of *--filter_threshold* was secondary, modulating performance within each *--mod_threshold,* with increased stringency generally reducing the F1 score.

We next evaluated whether ModkitOpt remains effective when validated sites originate from a different biological context than the nanopore dataset. Although the validated sites may not fully overlap with the sample-specific modification landscape, we hypothesised that a shared subset of true sites would suffice for parameter optimisation. Using nanopore datasets from HepG2 and K562 cells[4], and comparing against the original validated sites, we evaluated the same 36 threshold combinations and 1,000 stoichiometry cutoffs. For both m6A and pseU, the optimal thresholds were the same as previously identified using matched cell lines (**Supplementary Table 1**), and the relative performance across the parameter space was mostly conserved (**Supplementary Figures 1-4**).

We also considered how biological conclusions changed following threshold optimisation. Overall, the default strategy consistently distorted the number of detected sites, with optimised thresholds rescuing validated sites missed by the default, while also reducing the number of unvalidated sites (**Figure 2c,d**). This trend was consistent across all three cell lines tested for m6A, whereby unvalidated sites decreased by 49% (46-51%) on average, while 42% (33-46%) more validated sites were recovered, on average, compared to the default (**Supplementary Table 2**). These same trends were recapitulated for pseU in K562 and HepG2 cells, with

optimised thresholds recovering 13-fold more validated sites and reducing unvalidated site counts by 10% (3-17%) on average (**Supplementary Table 3**). Interestingly, threshold optimisation recovered 22-fold more validated pseU sites in HeLa cells, although the unvalidated site count was also inflated 2-fold. This is consistent with the default heuristic threshold being too stringent for this dataset, resulting in the vast majority of true modified sites being discarded.

Finally, we analysed the computational performance of ModkitOpt. When executed on an HPC system (PBSPro Scheduler) using default resource settings (**Methods**), ModkitOpt completed in 22 minutes for a 7GB modBAM file and 33 minutes for a 21GB modBAM file (**Supplementary Table 4**), demonstrating that comprehensive parameter optimisation can be performed efficiently on typical nanopore datasets.

Accurate quantification of RNA modifications is essential for extracting reliable biological insights from nanopore direct RNA sequencing. We demonstrate that the default heuristic employed by the de facto standard, modkit, systematically distorts RNA modification landscapes, which ModkitOpt rescues through comprehensive parameter optimisation by evaluating modification calls against validated sites. ModkitOpt reliably optimises modkit parameters, even when the biological context of the validated sites differs from that of the sample, provided that a subset of modification sites is shared, thus supporting its broad applicability for re-evaluating published datasets, as well as future studies. Implemented as an efficient Nextflow workflow, ModkitOpt integrates seamlessly into existing nanopore analysis pipelines to establish a new community standard for accurate and reliable reporting of RNA modifications across diverse experimental settings. ModkitOpt is available at https://github.com/comprna/modkitopt.

Methods

ModkitOpt workflow

ModkitOpt (**Figure 1**) takes as input a transcriptome-aligned modBAM[8] file generated by dorado, a reference transcriptome (FASTA), gene annotation (GTF or GFF3), and a specified modification type (e.g., m6A, pseU, m5C, inosine). Input modBAM files are filtered to retain primary alignments only, then sorted and indexed using samtools (v1.21)[9]. Modkit (v0.6.0) pileup is subsequently executed across a predefined grid of filtering parameters, spanning all combinations of *--filter-threshold* and *--mod-threshold* values (0.5, 0.75, 0.9, 0.95, 0.99, 0.995), producing transcriptome-coordinate bedMethyl files with site-level stoichiometry estimates. We also allow advanced users to optionally specify the --filter-threshold and --mod-threshold values to test.

Predicted sites are converted to genomic coordinates for comparison with validated sites. When multiple transcriptomic positions map to the same genomic site, stoichiometry values are aggregated using a weighted average across transcripts. ModkitOpt evaluates 1,000 candidate stoichiometry cutoffs uniformly spaced between 0 and 0.999, retaining sites with estimated stoichiometry exceeding each cutoff.

Precision (fraction of the predicted sites that are in the validated set), recall (fraction of the validated sites that are predicted) and F1 score ($2 \cdot \frac{(Pecision \cdot Recall)}{(Pecision + Recall)}$) are calculated for every

combination of modkit parameters and stoichiometry cutoff. For each parameter combination, the stoichiometry cutoff that maximises the F1 score is identified, and performance across all parameter combinations is compared to determine the combination of *--filter-threshold, --mod-threshold,* and stoichiometry cutoff that achieves the highest F1 score for the given dataset.

Validated sites are provided for mammalian m6A and pseU, derived from GLORI-identified m6A sites in HEK293T cells[5] and BACS-identified pseU sites in HeLa cells[6], respectively. ModkitOpt is distributed as a Nextflow workflow, enabling reproducible execution on local machines or HPC systems, with resource allocation automatically managed for efficient and scalable analysis. The default resource settings allocated by ModkitOpt are 8 CPUs and 8GB RAM for samtools filter and samtools sort; 8 CPUs and 30GB RAM for modkit pileup; 4 CPUs and 8GB RAM for samtools index; 1 CPU, 30GB RAM and 5GB job filesystem disk space for converting transcriptomic to genomic coordinates; and 1 CPU and 8GB RAM for all other tasks.

ModkitOpt outputs

ModkitOpt reports the optimal *--filter-threshold*, *--mod-threshold* and stoichiometry cutoff to the user and produces a bar plot depicting the precision, recall and F1 score across the parameter space (**Figure 1b**)[10]. The bar plot's data is also written in .tsv format. More detailed plots are additionally provided; a scatter plot visualising the F1 score across all combinations of *--mod-threshold* and *--filter-threshold,* using the optimal stoichiometry cutoff per threshold combination (**Supplementary Figure 5**); and the set of precision-recall curves for each threshold combination, annotated with the maximum F1 score per curve (**Supplementary Figure 6**).

References

1. Garalde, D. R. *et al.* Highly parallel direct RN A sequencing on an array of nanopores. *Nat Methods* 15, 201–206 (2018).
2. nanoporetech/dorado: Oxford Nanopore's Basecaller. https://github.com/nanoporetech/dorado.
3. nanoporetech/modkit: A bioinformatics tool for working with modified bases. https://github.com/nanoporetech/modkit.
4. Zou, Y. *et al.* A comparative evaluation of computational models for RNA modification detection using nanopore sequencing with RNA004 chemistry. *Brief Bioinform* 26, (2025).
5. Liu, C. *et al.* Absolute quantification of single-base m6A methylation in the mammalian transcriptome using GLORI. *Nat Biotechnol* 41, 355–366 (2023).
6. Xu, H. *et al.* Absolute quantitative and base-resolution sequencing reveals comprehensive landscape of pseudouridine across the human transcriptome. *Nat Methods* 21, 2024–2033 (2024).
7. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* 34, 3094–3100 (2018).

8. Li, H. *et al*. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* 25, 2078–2079 (2009).

9. Wickham, H. *et al*. Welcome to the Tidyverse. *J Open Source Softw* 4, 1686 (2019).

Funding

This research was supported by the Australian Research Council (ARC) Discovery Project grants DP250100070 and DP250103133; by the National Health and Medical Research Council (NHMRC) Ideas Grant 2018833; by a PhD scholarship and an Innovator Grant from the ANU TALO Computational Biology Talent Accelerator, and by the ARC Centre of Excellence for the Mathematical Analysis of Cell Systems (MACSYS) (CE230100001). This research was also indirectly supported by the Australian Government's National Collaborative Research Infrastructure Strategy (NCRIS) through access to computational resources provided by the National Computational Infrastructure (NCI) through the National Computational Merit Allocation Scheme (NCMAS) and the ANU Merit Allocation Scheme (ANUMAS). The funding bodies had no role in study design, data collection, or data analysis.

Author Contributions

AS, SP and EE conceptualised the study. AS led the software development and data analysis, with essential contributions from SP & EE. AS wrote the manuscript, and SP and EE reviewed the manuscript.

Competing Interests

AS has previously received funds from Oxford Nanopore Technologies for conference travel.