The Journal of Open Source Software

perbase: A performant per-base sequencing metrics toolkit with accurate handling of complex alignments

Seth Stadick [1]

1 Bio-Rad Laboratories

DOI: 10.21105/joss.09774

Software

- Review
- Repository
- Archive

Editor: Frederick Boehm

Reviewers:

- @mbhall88
- @chrisamiller

Submitted: 29 July 2025
Published: 19 February 2026

Summary

perbase is a command-line toolkit for calculating per-base sequencing metrics from alignment files (BAM/CRAM). The primary tool, base-depth, provides comprehensive nucleotide-level information including depth, base composition, insertions, deletions, and quality metrics at each genomic position. Built with Rust's concurrency system, perbase delivers performant processing of high-throughput sequencing data while maintaining correctness in complex genomic contexts such as overlapping mate pairs, deletions, and reference skips.

Statement of need

Per-base sequencing metrics are fundamental to genomic analyses, from variant calling to coverage assessment. Existing tools like `sambamba depth` (Tarasov et al., 2015), `samtools depth` (Li et al., 2009), and `bam-readcount` (Khanna et al., 2022) provide similar functionality but may differ in their handling of specific alignment features. As sequencing datasets grow larger, there is a need for tools that combine performance with correct handling of edge cases.

For instance, tools differ in how they calculate depth: perbase counts deletions (D in CIGAR) toward depth while `bam-readcount` does not; perbase correctly excludes reference skips (N in CIGAR) from depth while sambamba includes them. These distinctions matter for downstream analyses where accurate depth representation affects variant calling and coverage assessment.

Software Design

perbase is implemented in Rust and uses a multi-threaded architecture where genomic regions are processed in parallel. The toolkit automatically scales with available CPU cores while maintaining bounded memory usage through configurable chunk sizes and message passing buffers.

Core Features of base-depth

The base-depth tool walks over every position in the BAM/CRAM file and calculates:

- **Depth**: Total count of the nucleotides plus deletions at each position
- **Base composition**: Individual counts for A, C, G, T, N, and IUPAC ambiguity codes (R, Y, S, W, K, M)
- **Insertions**: Count of insertions starting to the right of each position
- **Deletions**: Count of deletions covering each position (included in depth)
- **Reference skips**: Count of reference skip operations (not included in depth)
- **Failed reads**: Count of reads failing user-specified filters at each position
- **Quality filtering**: Bases below a minimum quality threshold are counted as N

- **Near max depth flag**: Identifies positions within 1% of the specified maximum depth

Mate-Pair Overlap Resolution

When the `--mate-fix` flag is enabled, `perbase base-depth` resolves overlapping mate pairs to prevent double-counting while preserving the highest-confidence base calls. The tool offers nine different resolution strategies, selectable via `--mate-resolution-strategy`:

Quality-based strategies: - **BaseQualMapQualFirstInPair**: Prioritizes base quality, then MAPQ, then first mate - **MapQualBaseQualFirstInPair**: Prioritizes MAPQ, then base quality, then first mate - **Original** (default): Simple MAPQ-based selection, choosing first mate for ties

Ambiguity-preserving strategies: - **BaseQualMapQualIUPAC**: Base quality prioritized, returns IUPAC codes for ties - **MapQualBaseQualIUPAC**: MAPQ prioritized, returns IUPAC codes for ties - **IUPAC**: Ignores quality scores, always returns IUPAC codes for different bases

Conservative strategies: - **BaseQualMapQualN**: Base quality prioritized, returns N for ties if bases are ambiguous - **MapQualBaseQualN**: MAPQ prioritized, returns N for ties if bases are ambiguous - **N**: Most conservative, returns N for any base differences

All strategies first check user-based read filters. If one mate fails filters, the other is chosen. If both fail, the first mate is chosen by default. For reads that are deletions, reference skips, or lack a base call, all strategies fall back to the Original strategy (MAPQ $\rightarrow$ first in pair).

When IUPAC strategies encounter different bases, they return standardized ambiguity codes: R (puRine: A/G), Y (pYrimidine: C/T), S (Strong: G/C), W (Weak: A/T), K (Keto: G/T), M (aMino: A/C). Identical bases return themselves (e.g., A+A$\rightarrow$A), and any other combinations return N.

These strategies provide fine-grained control over overlap resolution, allowing users to optimize for their specific analysis requirements: use BaseQual strategies when base quality is most reliable, MapQual strategies in repetitive regions where mapping confidence matters more, IUPAC variants to preserve ambiguity information for downstream analysis, N variants for conservative base calling, and FirstInPair variants for deterministic results without ambiguity codes.

Output Format

The tool produces a tab-separated output with the following columns:

Column	Description
REF	Reference sequence name
POS	Position on the reference sequence
REF_BASE	Reference base at the position (if reference supplied)
DEPTH	Total depth: SUM(A, C, G, T, N, R, Y, S, W, K, M, DEL)
A, C, G, T, N	Count of each standard nucleotide
R, Y, S, W, K, M	Count of IUPAC ambiguity codes (when using IUPAC strategies)
INS	Insertions starting after this position
DEL	Deletions covering this position
REF_SKIP	Reference skips covering this position
FAIL	Reads failing filters at this position
NEAR_MAX_DEPTH	Flag if position is within 1% of max depth

Additional Tools

only-depth: Provides rapid depth-only calculations. The design, inspired by `mosdepth` (Pedersen & Quinlan, 2018), merges adjacent positions with identical depth to reduce output size. A `--fast-mode` option calculates depth using only read start/stop positions for maximum speed.

merge-adjacent: A utility for merging adjacent intervals with the same depth value, useful for creating compact coverage representations.

Performance Evaluation

To demonstrate performance, we benchmark `base-depth` against `sambamba` (Tarasov et al., 2015) on a 30X whole genome sequencing dataset (HG00157 from the 1000 Genomes Project (The 1000 Genomes Project Consortium, 2015), (Byrska-Bishop et al., 2022)). The benchmark script processes the full genome and measures runtime and memory usage using hyperfine (Peter, 2023), a command-line benchmarking tool that performs multiple runs and provides statistical analysis. Benchmarks were performed on a system with an AMD Ryzen 9 3950X 16-Core Processor (32 threads) and 64 GB of RAM. Both tools used 32 threads for processing.

The following commands were used for benchmarking:

```
# Standard mode
perbase base-depth -t 32 -o output.tsv input.bam
sambamba depth base -t 32 -F "" -o output.tsv input.bam

# Mate-fix mode
perbase base-depth -t 32 -m -o output.tsv input.bam
sambamba depth base -t 32 -F "" -m -o output.tsv input.bam
```

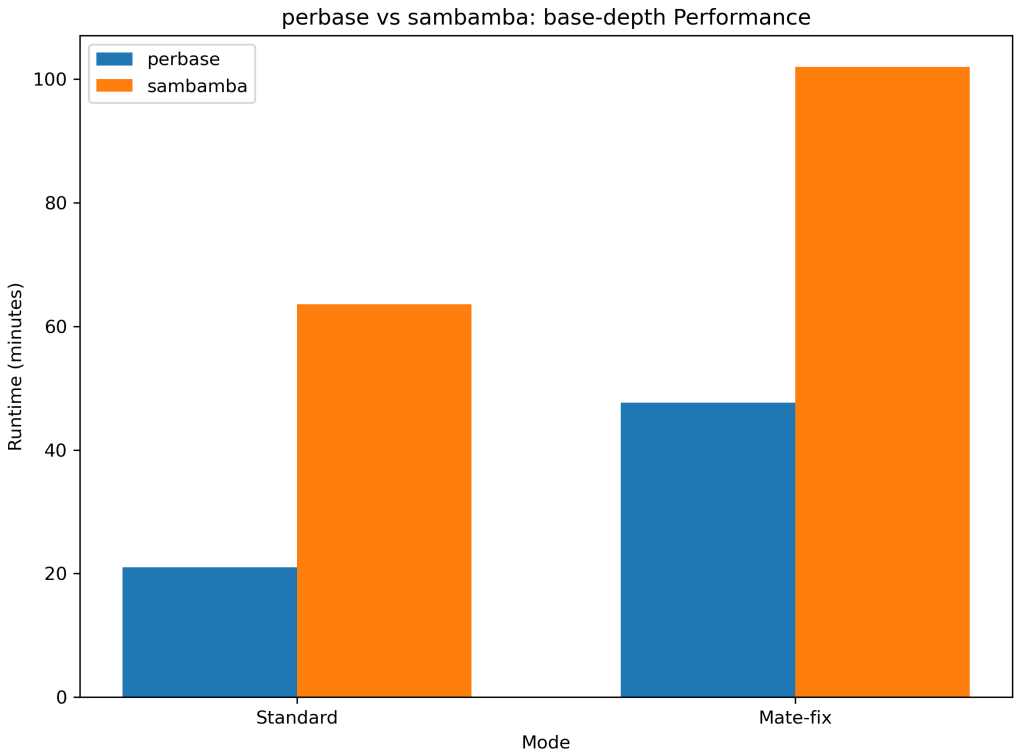

Figure 1: Performance comparison between perbase and sambamba showing runtime in minutes for both standard and mate-fix modes. perbase demonstrates 3.0x faster performance in standard mode and 2.1x faster performance in mate-fix mode (using BaseQualMapQualFirstInPair strategy).

The results show that perbase significantly outperforms sambamba in both standard and mate-fix modes, with speed improvements of 3.0x and 2.1x respectively. Performance across the nine mate-fix resolution strategies is remarkably consistent, with all methods completing within a 0.5-minute range (47.7-48.2 minutes), indicating that the algorithmic complexity of different resolution strategies has minimal impact on overall runtime.

Mate-fix Strategy	Runtime (minutes)	Relative to Fastest
BaseQualMapQualFirstInPair	47.7	1.00x
N	47.7	1.00x
Original	47.7	1.00x
BaseQualMapQualN	47.8	1.00x
MapQualBaseQualN	47.9	1.00x
IUPAC	48.0	1.01x
BaseQualMapQualIUPAC	48.0	1.01x
MapQualBaseQualIUPAC	48.2	1.01x
MapQualBaseQualFirstInPair	48.2	1.01x

This performance advantage is achieved through efficient parallelization and optimized memory access patterns.

Research Impact Statement

perbase has accumulated over 49,000 downloads from Bioconda and serves as a foundational dependency for downstream tools including pbr, which integrates perbase pileups with lua

expressions. Community engagement is demonstrated through external issue reports and feature requests. The repository includes reproducible benchmark scripts and uses standardized 1000 Genomes Project test data to facilitate independent validation.

AI usage disclosure

AI tools were used during development: GitHub Copilot for code review suggestions, and Claude Code for some issue triage, some test writing, and manuscript proofreading. All outputs were reviewed and validated by the author, who made all core design and architectural decisions.

Availability and Installation

`perbase` is available through multiple channels: - Conda: `conda install -c bioconda perbase` - Cargo: `cargo install perbase` - Pre-compiled binaries from GitHub releases: https://github.com/sstadick/perbase/releases

Source code is available at https://github.com/sstadick/perbase under the MIT license.

Acknowledgements

We acknowledge the Rust bioinformatics community and the authors of key dependencies including rust-htslib.

References

Byrska-Bishop, M., Evani, U. S., Zhao, X., Basile, A. O., Abel, H. J., Regier, A. A., Corvelo, A., Clarke, W. E., Musunuri, R., Nagulapalli, K., Fairley, S., Runnels, A., Winterkorn, L., Lowy, E., The 1000 Genomes Project Consortium, Flicek, P., Germer, S., Brand, H., Hall, I. M., … Zody, M. C. (2022). High-coverage whole-genome sequencing of the expanded 1000 genomes project cohort including 602 trios. *Cell*, *185*(18), 3426–3440.e19. https://doi.org/10.1016/j.cell.2022.08.004

Khanna, A., Larson, D. E., Srivatsan, S., Mosior, M., Abbott, T. E., Kiwala, S., Ley, T. J., Duncavage, E. J., Walter, M. J., Walker, J. R., Miller, C. A., McMichael, J. F., Griffith, O. L., & Griffith, M. (2022). Bam-readcount - rapid generation of basepair-resolution sequence metrics. *Journal of Open Source Software*, *7*(69), 3722. https://doi.org/10.21105/joss.03722

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., & 1000 Genome Project Data Processing Subgroup. (2009). The sequence alignment/map format and SAMtools. *Bioinformatics*, *25*(16), 2078–2079. https://doi.org/10.1093/bioinformatics/btp352

Pedersen, B. S., & Quinlan, A. R. (2018). Mosdepth: Quick coverage calculation for genomes and exomes. *Bioinformatics*, *34*(5), 867–868. https://doi.org/10.1093/bioinformatics/btx699

Peter, D. (2023). *Hyperfine: A command-line benchmarking tool* (Version 1.16.1). https://github.com/sharkdp/hyperfine

Tarasov, A., Vilella, A. J., Cuppen, E., Nijman, I. J., & Prins, P. (2015). Sambamba: Fast processing of NGS alignment formats. *Bioinformatics*, *31*(12), 2032–2034. https://doi.org/10.1093/bioinformatics/btv098

The 1000 Genomes Project Consortium. (2015). A global reference for human genetic variation. *Nature*, *526*(7571), 68–74. https://doi.org/10.1038/nature15393