

Correcting PCR amplification errors in unique molecular identifiers to generate accurate numbers of sequencing molecules

Received: 8 April 2023

Accepted: 4 January 2024

Published online: 05 February 2024

 Check for updates

Jianfeng Sun¹, Martin Philpott¹, Danson Loi¹, Shuang Li², Pablo Monteagudo-Mesas², Gabriela Hoffman³, Jonathan Robson³, Neelam Mehta¹, Vicki Gamble¹, Tom Brown Jr³, Tom Brown Sr⁴, Stefan Canzar^{5,6}, Udo Oppermann^{1,7} & Adam P. Cribbs^{1,7} ✉

Unique molecular identifiers are random oligonucleotide sequences that remove PCR amplification biases. However, the impact that PCR associated sequencing errors have on the accuracy of generating absolute counts of RNA molecules is underappreciated. We show that PCR errors are a source of inaccuracy in both bulk and single-cell sequencing data, and synthesizing unique molecular identifiers using homotrimeric nucleotide blocks provides an error-correcting solution that allows absolute counting of sequenced molecules.

Unique molecular identifiers (UMIs)¹ distinguish molecules in sequencing, enabling correction for biases in sampling and PCR amplification across next-generation and third-generation sequencing methods, including bulk RNA^{2,3}, single-cell RNA^{4,5} and DNA approaches^{6–8}. However, the accuracy of molecular quantification can be affected by the varying sequencing quality of different platforms⁹. Different sequencing platforms necessitate varied PCR cycling conditions, potentially introducing UMI errors that may result in inaccurate molecule counts (Supplementary Fig. 1). Unlike sample barcodes for multiplexing or cell barcodes in single-cell sequencing, which can be whitelisted due to a limited pool of barcodes¹⁰, UMIs cannot be corrected using this approach as their synthesis is random. Therefore, UMIs are often corrected using computational approaches¹¹, concatemeric consensus sequencing¹² or by bespoke UMI designs^{13,14}. Despite several computational approaches that leverage Hamming distances^{15,16}, graph networks^{11,13} or thresholding on UMI frequency⁴, experimental validation of these solutions is lacking, with simulations indicating persistent UMI errors postcomputational demultiplexing¹³.

We reasoned that using homotrimer nucleotides to synthesize UMIs would simplify error detection and correction by using a ‘majority

vote’ method (Fig. 1a and Supplementary Fig. 2). Our method labels RNA with homotrimeric UMIs at either end for enhanced error detection and indel tolerance, compatible with the ONT (Oxford Nanopore Technologies), PacBio or Illumina platforms (Fig. 1a,b). UMIs are processed by assessing trimer nucleotide similarity; errors are corrected by adopting the most frequent nucleotide in a majority vote approach (Fig. 1c). Our simulations reveal that a demultiplexing strategy incorporating homotrimers, along with a set coverage approach, outperforms the existing gold standard of monomer-based UMI-tools demultiplexing (Supplementary Fig. 3). By synergistically integrating homotrimeric correction and set coverage techniques, our method achieves a substantial improvement in the detection and recovery of simulated UMIs. This optimized performance exceeds the results obtained when relying solely on a homotrimer majority vote approach (Supplementary Fig. 4).

While sequencing simulations can offer valuable insights, their real-world applicability may be limited by biases. To validate our homotrimer UMI error correction approach, we conducted experiments using a common molecular identifier (CMI) attached to every captured RNA molecule (Supplementary Fig. 5). Having the same molecule attached to every RNA guarantees that, in the absence of errors, each

¹Botnar Research Centre, Nuffield Department of Orthopaedics, Rheumatology and Musculoskeletal Sciences, National Institute of Health Research Oxford Biomedical Research Unit (BRU), University of Oxford, Oxford, UK. ²Gene Center, Ludwig-Maximilians University of Munich, Munich, Germany.

³ATDBio Ltd (now part of Biotage), Magdalen Centre, Oxford Science Park, Oxford, UK. ⁴Chemistry Research Laboratory, Department of Chemistry, University of Oxford, Oxford, UK. ⁵Department of Computer Science and Engineering, The Pennsylvania State University, University Park, PA, USA. ⁶Huck Institutes of the Life Sciences, The Pennsylvania State University, University Park, PA, USA. ⁷Oxford Centre for Translational Myeloma Research, University of Oxford, Oxford, UK. ✉e-mail: adam.cribbs@ndorms.ox.ac.uk

transcript is only counted once. However, if errors are introduced into the CMI, transcripts will be overcounted. This provides a means for assessing the accuracy of library preparation and sequencing, as well as the impact of errors on the transcript counts (Supplementary Fig. 5b).

We attached the CMI to equimolar concentrations of mouse and human complementary DNA (cDNA) at the 3' end, PCR amplified and then split the sample for sequencing on Illumina, PacBio or ONT platforms. We calculated the Hamming distance between the observed and expected CMI sequence to measure sequencing accuracy. Our results show that 73.36, 68.08 and 89.95% of CMIs were correctly called using Illumina, PacBio and the latest kit14 ONT chemistry, respectively (Fig. 1d and Supplementary Figs. 6–8). Older ONT chemistry gave substantially lower accuracy (Supplementary Fig. 9), but the use of super accuracy base calling led to substantial improvements (Supplementary Fig. 10). Using our homotrimeric error correction approach, we were able to correctly call 98.45, 99.64 and 99.03% of CMIs for Illumina, PacBio and the latest ONT chemistry, respectively (Fig. 1d). We hypothesized that the lower accuracy of Illumina and PacBio, when compared to ONT sequencing, may be due to the use of polymerases that are integral to the sequencing process (for example, bridge amplification¹⁷ and circular consensus sequencing¹⁸ with Illumina and PacBio, respectively). To discern sequencing and PCR errors, we amplified a CMI-tagged cDNA library with increasing PCR cycles and sequenced using ONT's Minion. Trimer barcodes added during PCR allowed for batch effect minimization and independent sequencing accuracy assessment. High barcode accuracy was noted, with homotrimer correction offering negligible improvement (Fig. 1e). Based on these results, it can be inferred that sequencing errors make a negligible contribution to the overall error rate. However, we observed a substantial increase in the number of errors within our CMIs with increasing PCR cycles (Fig. 1f and Supplementary Fig. 11). Our homotrimer approach was able to correct a significant proportion of errors observed within the CMIs (Fig. 1g). This suggests that PCR can be a significant source of UMI error. We also benchmarked homotrimer error correction against both UMI-tools¹¹ and TRUmiCount¹⁹ and found substantial improvements in error correction (Fig. 1h and Supplementary Fig. 12). We observed minimal indel errors, suggesting that most errors were substitutions (Supplementary Fig. 12). It is important to highlight that monomer UMIs using Hamming distance, such as those in UMI-tools and TRUmiCount, cannot correct indel errors due to the potential for a single indel to inflate the Hamming distance beyond correctability. Our methodology overcomes this by including indel correction.

We next conducted an experiment to correct PCR errors using homotrimers and treated Ewing's RM82 sarcoma cells with a CLK1 splicing kinase inhibitor. This induced splicing perturbations, allowing observation of an exaggerated differential transcript effect, followed by ONT or Illumina sequencing (Fig. 1i–l and Supplementary Figs. 13

and 14). When we compared monomer UMI correction to our homotrimer correction methodology, we found differences in the number of differentially expressed genes and transcripts between splicing inhibition and control conditions. Specifically, for genes and transcripts, we observed discordant rates of 7.8 and 11%, respectively (Fig. 1i–j and Supplementary Tables 2 and 3). The discordance rate indicates exclusive gene or transcript regulation in one condition over another. More genes were differentially regulated after monomer UMI-tools correction than homotrimer correction, exemplified by read counts for TLE5 and FRG2 genes postcorrection (Fig. 1k and Supplementary Fig. 15). In addition, the homotrimer correction approach led to an increased fold enrichment of genes associated with gene ontology terms related to DNA replication and splicing (Supplementary Fig. 16), highlighting the improved accuracy of our method in identifying biologically relevant gene sets. Additionally, we also observed 4.7% discordant differentially expressed genes between UMI-tools and homotrimer correction following Illumina sequencing (Supplementary Fig. 13c,d).

To understand the effect of PCR errors on single-cell sequencing accuracy, we used the 10X Chromium system with monomer UMIs to encapsulate JYN3 human and 5TGM1 mouse cells, followed by ten PCR cycles. Subsequently, we divided the PCR product into two portions and performed additional PCR amplification, resulting in a combined number of PCR cycles of 20 or 25. We then prepared and sequenced these libraries using ONT's PromethION platform and after assigning cell barcodes (Fig. 2a) and filtering, clustering and annotating the cells (Fig. 2b), we observed that the library subjected to 25 cycles of PCR had a greater number of UMIs compared to the library that underwent 20 PCR cycles (Fig. 2c and Supplementary Fig. 17). This suggests that PCR errors contribute to inaccurate counting of transcripts and an inflated UMI count. We next performed differential gene expression and identified 50 differentially expressed transcripts (Supplementary Tables 4 and 5). For example, transcripts ENSMUST00000034966 (Fig. 2d; Rpl4, ribosomal protein L4) and ENST00000532223 (Fig. 2e; IGLL5, immunoglobulin lambda), were identified as significant ($P_{\text{adj}} < 0.05$) in this differential expression analysis, highlighting the contribution of PCR errors to inaccurate transcript counting.

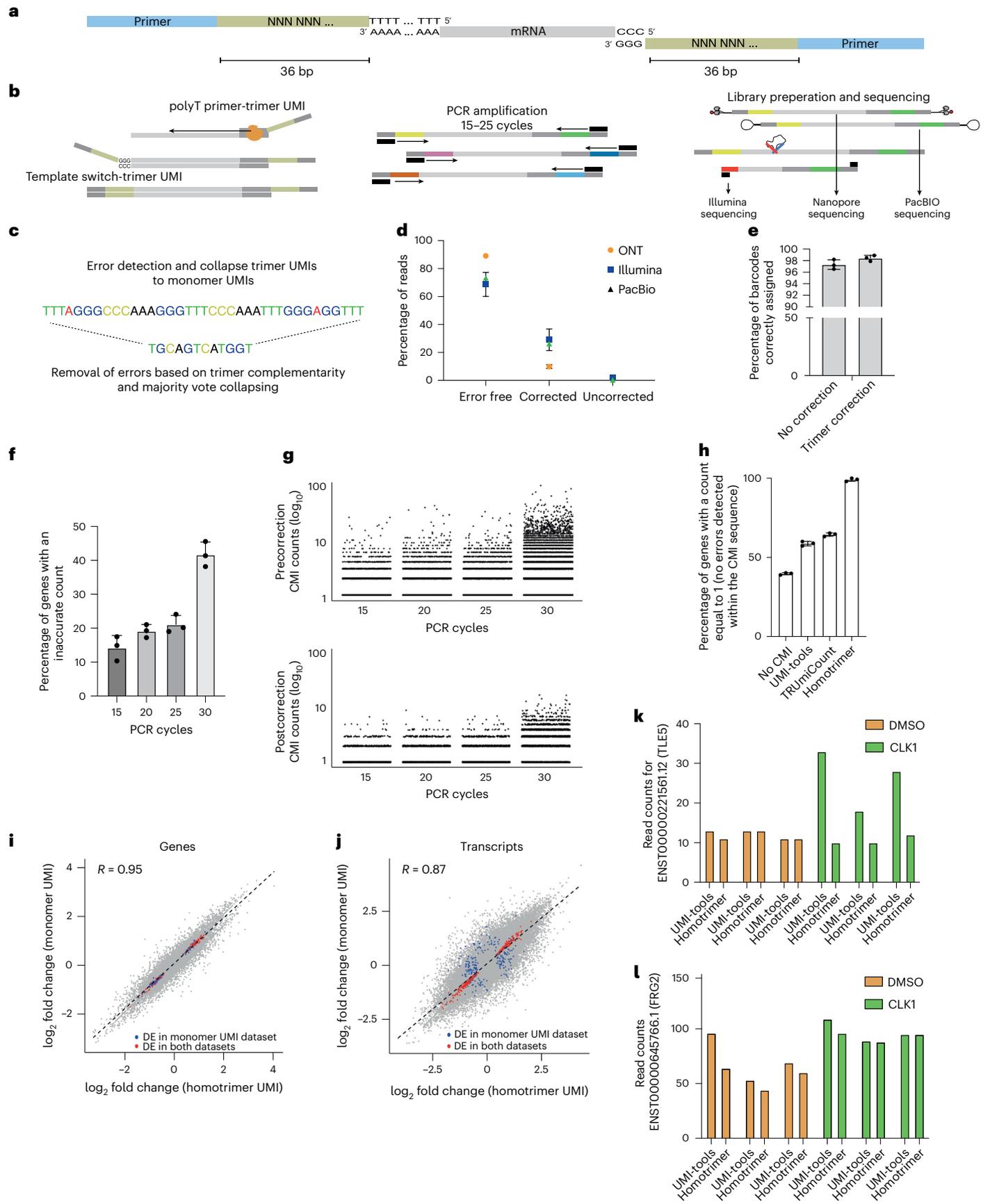
Next, we encapsulated JYN3 human and 5TGM1 mouse cells using Drop-seq²⁰ with trimer barcoded beads, conducted reverse transcription and template switching with a CMI and initiated ten PCR cycles. The PCR product was split into four aliquots for further amplification to 20, 25, 30 and 35 PCR cycles, respectively, before sequencing on the ONT Minion platform. Our results indicate a decrease in the percentage of reads with accurate CMIs as the number of PCR cycles increases. We show that homotrimer correction leads to 96–100% correction of CMI sequences (Fig. 2g and Supplementary Fig. 18). This underscores the effectiveness of this approach in removing errors introduced by PCR. Subsequently, we sequenced the libraries that underwent 20 or 25 PCR

Fig. 1 | Enhanced accuracy in bulk mRNA sequencing using homotrimer UMI-based approach to mitigate PCR-induced errors. **a**, A schematic showing attachment of 3' and 5' UMIs to mRNA. **b**, A schematic showing the homotrimeric UMI approach. **c**, Errors are then corrected using the homotrimer correction method. **d**, Percentage of CMIs that are correctly sequenced and then error corrected using homotrimer correction across Illumina, PacBio and ONT sequencing platforms. Experiments for Illumina and ONT were performed in triplicate, whereas PacBio sequencing was conducted as a single run. Parameters for simulations: sequencing error rate 0.001, length of UMI 8, PCR cycles 10 and PCR error rate 0.000001. **e**, Barcode assignment using homotrimer barcodes before and after majority vote correction. **f**, Percentage of genes with an accurate CMI count following increased PCR cycles of the same sequencing library. Data shown in the figure are from one single run. **g**, \log_{10} CMI counts plotted for each transcript pre- and postmajority vote correction. Each dot represents an individual transcript (the ground truth count for each transcript should be equal to 1, any counts above this are indicative of an error). The data in this figure are representative of one sample in **f**. **h**, Percentage of genes with an accurate

CMI count following 20 PCR cycles then using ONT sequencing and counting using UMI-tools, TRUmiCount correction and homotrimer error correction. **i–l**, RM82 sarcoma cells were treated with DMSO or SGC-CLK-1 for 24 hours and then sequenced using the PromethION platform. **i, j**, Scatter plot of the \log_2 fold changes obtained from randomly collapsing each sequenced trimer UMI and then applying UMI-tools deduplication versus the \log_2 fold changes obtained from homotrimer UMI correction and counting for genes (**i**) and transcripts (**j**). Red points indicate the overlapping significant genes and/or transcripts and blue points indicate genes and/or transcripts that were discordantly significantly differentially expressed. DE, differential expression. **k**, TLE5 transcript read counts showing the expression for DMSO and SGC-CLK1 following the application of UMI-tools or homotrimer correction. **l**, FRG2 transcript read counts showing the expression for DMSO and SGC-CLK1 following the application of UMI-tools or homotrimer correction. For **k** and **l**, three replicates are shown for each condition. **d, e, f** and **h**, Error bars represent standard deviation (s.d.) from three independent experiments.

cycles using ONT's PromethION platform (Supplementary Fig. 19). Our results show that by incorporating homotrimers within the barcode region an increase, albeit low (~15%), in the numbers of cells recovered

was achieved (Fig. 2h). Monomeric UMI deduplication resulted in over 300 differentially regulated transcripts between the 20 and 25 cycle libraries (Fig. 2i and Supplementary Table 4). On the contrary, the



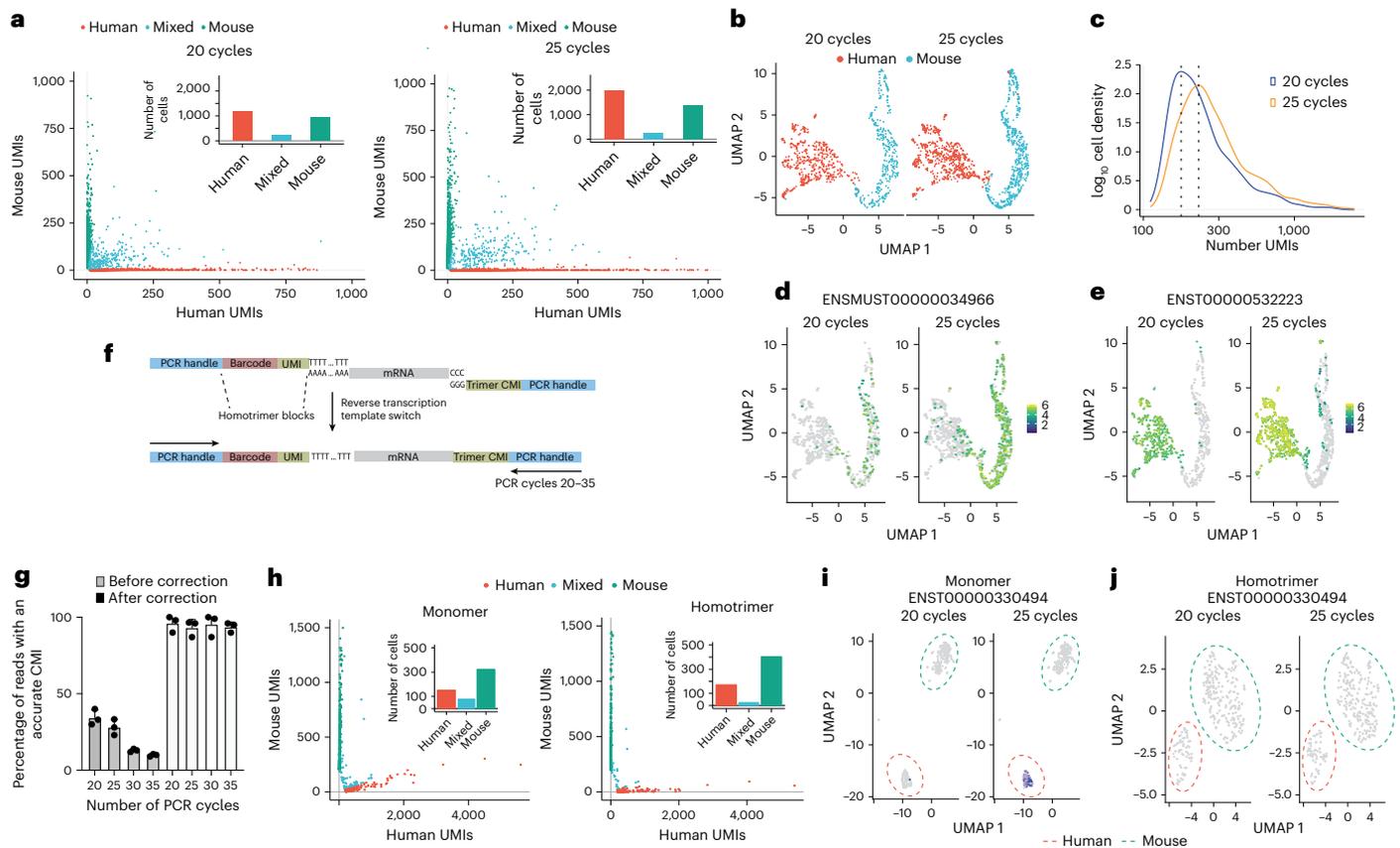


Fig. 2 | Homotrimer UMIs enhance differential expression accuracy in single-cell sequencing. **a**, Human Jurkat and mouse 5TGM1 cells were mixed for encapsulation and cDNA synthesis using 10X chromium followed by nanopore sequencing. Each dot represents an individual cell. **b**, A UMAP of 10X chromium data showing the integration, clustering and annotation of human and mouse cells following 20 and 25 cycles of PCR. **c**, A density plot for the 10X chromium data showing the \log_{10} density of the number of UMIs following 20 and 25 cycles of PCR. The dotted line shows the maximum density for each condition. **d**, A UMAP showing the expression of ENSMUST00000034966 between libraries amplified following 20 and 25 PCR cycles. **e**, A UMAP showing the expression of ENST00000532223 between libraries amplified following 20 and 25 cycles of PCR. **f**, A schematic showing the homotrimer UMI drop-seq library preparation

approach and template switching attachment of a homotrimer CMI to single-cell captured mRNAs. **g**, Drop-seq libraries were sequenced using the Flongle sequencing device, graphs show the percentage of reads that have an accurate CMI following amplification of the same library using 20, 25, 30 and 35 cycles of PCR before and after homotrimer correction. Error bars are the s.d. of three independent experiments. **h**, Barnyard plots showing the expression of mouse and human cells following 20 and 25 cycles of PCR and sequencing using a PromethION sequencing device. Each dot represents an individual cell. **i, j**, UMAP plots showing the transcript expression of ENST00000330494 following monomer-based UMI-tools demultiplexing (**i**) and homotrimer-based demultiplexing (**j**). In each UMAP plot, each dot represents a cell. Data shown in **g** were collected in a single sequencing run and $n = 1$.

application of homotrimer correction did not reveal any significantly differentially regulated transcripts. Transcripts with high counts after 25 cycles of monomer UMI correction saw a reduction when subjected to homotrimer UMI correction (Fig. 2j), demonstrating the robustness of homotrimer UMIs to remove errors.

Our research highlights the importance of accurate UMI quantification in sequencing, endorsing homotrimer UMIs to improve read count precision. Homotrimers notably mitigate PCR-induced UMI errors, optimizing molecule count sequencing. Although they extend the oligonucleotide length, their suitability for long-read sequencing—unrestricted by read length—outweighs this limitation, offering substantial benefits for studies requiring rigorous UMI quantification.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at <https://doi.org/10.1038/s41592-024-02168-y>.

References

- Hug, H. & Schuler, R. Measurement of the number of molecules of a single mRNA species in a complex mRNA preparation. *J. Theor. Biol.* **221**, 615–624 (2003).
- Aird, D. et al. Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. *Genome Biol.* **12**, R18 (2011).
- Kivioja, T. et al. Counting absolute numbers of molecules using unique molecular identifiers. *Nat. Methods* **9**, 72–74 (2011).
- Islam, S. et al. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nat. Methods* **11**, 163–166 (2014).
- Hagemann-Jensen, M. et al. Single-cell RNA counting at allele and isoform resolution using Smart-seq3. *Nat. Biotechnol.* **38**, 708–714 (2020).
- Schmitt, M. W. et al. Detection of ultra-rare mutations by next-generation sequencing. *Proc. Natl Acad. Sci. USA* **109**, 14508–14513 (2012).
- Kukita, Y. et al. High-fidelity target sequencing of individual molecules identified using barcode sequences: de novo detection and absolute quantitation of mutations in plasma cell-free DNA from cancer patients. *DNA Res.* **22**, 269–277 (2015).

8. Peng, X. & Dorman, K. S. Accurate estimation of molecular counts from amplicon sequence data with unique molecular identifiers. *Bioinformatics* **39**, btad002 (2023).
9. Shafin, K. et al. Haplotype-aware variant calling with PEPPER-Margin-DeepVariant enables high accuracy in nanopore long-reads. *Nat. Methods* **18**, 1322–1332 (2021).
10. You, Y. et al. Identification of cell barcodes from long-read single-cell RNA-seq with BLAZE. *Genome Biol.* **24**, 66 (2023).
11. Smith, T., Heger, A. & Sudbery, I. UMI-tools: modeling sequencing errors in Unique Molecular Identifiers to improve quantification accuracy. *Genome Res.* **27**, 491–499 (2017).
12. Volden, R. & Vollmers, C. Single-cell isoform analysis in human immune cells. *Genome Biol.* **23**, 47 (2022).
13. Philpott, M. et al. Nanopore sequencing of single-cell transcriptomes with scCOLOR-seq. *Nat. Biotechnol.* **39**, 1517–1520 (2021).
14. Karst, S. M. et al. High-accuracy long-read amplicon sequences using unique molecular identifiers with Nanopore or PacBio sequencing. *Nat. Methods* **18**, 165–169 (2021).
15. Tsagiopoulou, M. et al. UMIc: a preprocessing method for UMI deduplication and reads correction. *Front. Genet.* **12**, 660366 (2021).
16. Bose, S. et al. Scalable microfluidics for single-cell RNA printing and sequencing. *Genome Biol.* **16**, 120 (2015).
17. Shagin, D. A. et al. A high-throughput assay for quantitative measurement of PCR errors. *Sci. Rep.* **7**, 2718 (2017).
18. Potapov, V. & Ong, J. L. Examining sources of error in PCR by single-molecule sequencing. *PLoS ONE* **12**, e0169774 (2017).
19. Pflug, F. G. & von Haeseler, A. TRUmiCount: correctly counting absolute numbers of molecules using unique molecular identifiers. *Bioinformatics* **34**, 3137–3144 (2018).
20. Macosko, E. Z. et al. Highly parallel genome-wide expression profiling of individual cells using nanoliter droplets. *Cell* **161**, 1202–1214 (2015).

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2024

ONT bulk RNA-seq analysis

We performed base calling on the raw fast5 data to generate fastq files using Guppy (v.6.4.8) (`guppy_basecaller -compress-fastq -c [cfg file] -x 'cuda:0'`) in graphical processing unit (GPU) mode from ONT running on a RTX3090 graphics card. The fastq data were processed using a custom pipeline 'pipeline_count' written using `cgatcore`²² and included within the TallyTriN repository. Briefly, the quality of each fastq file was evaluated using the `fastqc` toolkit²³ and summary statistics were collated using `Multicq`²⁴. We then identified the polyA associated UMI sequence by searching for the polyA region and reverse complementing the read if it did not appear in the correct orientation. The 30 bp UMI was identified upstream of the SMART primer by pattern matching for `GTAAGTCTGCGTTGATACCAGTCTGCTT`. The set coverage method for removing homotrimer methods was then applied for UMI demultiplexing; if the UMI contained more than five errors then the read was removed. The demultiplexed UMI sequence was then added to the read name. Next, the template switch oligo (TSO)-associated UMI was identified using the SMART primer sequence `AAGCAGTGGTATCAACGCAGAGTAAT`. The 30 bp UMI sequence was first subjected to error correction or removed from the read, depending on the number of UMI errors detected. Subsequently, the corrected UMI, if applicable, was appended to the read name, enhancing the accuracy and use of the data. Both the TSO and polyA associated UMIs and primer sequences were removed from the read sequence. For transcript level analysis, the fastq file was then mapped against the transcriptome using `minimap2` (v.2.25) with the following settings: `-ax map-ont -p 0.9 --end-bonus 10 -N 3`. The resulting `.sam` file was then sorted and indexed using `samtools`²⁵. A custom script, in which `pysam` (v.0.21.0) was used to parse the output `.sam` file, was then used to add the transcript name to the XT tag of the `samfile` for downstream counting by homotrimer deduplication or UMI-tools. For gene level analysis, the fastq data were mapped using `minimap2` using the following setting: `-ax splice -k 14 --sam-hit-only --secondary=no --junc-bed`. The resulting `.sam` file was then sorted and indexed followed by feature annotation using `featurecounts` (v.2.0.1)²⁶ using the following settings to generate an annotated `.bam` file: `featureCounts -a (gtf) -o (output) -R BAM`. This `.bam` file was then used for downstream counting by UMI-tools or homotrimer correction. The reference transcriptome and genomes used for the analysis were `hg38_ensembl98` and `mm10_ensembl88`. The resulting count tables were then used for differential gene expression analysis, which was performed using `DESeq2` v.1.40.2 (ref. 27) within the R statistical framework v.4.3.1.

Illumina bulk RNA-seq analysis

The data were processed using a custom `cgatcore`²⁸ (v.0.6.15) written pipeline 'pipeline_illumina'. Briefly, the UMIs contained in read1 were corrected based on homotrimer complementarity or were removed from the analysis depending on a set error threshold. The paired fastq files were then mapped using `hisat2` (v.2.2.1)²⁹ before features being counted using `featureCounts`³⁰ (v.2.0.3) using the following commands: `featureCounts -a (gtf) -o (output) -R BAM`. The resulting XT tagged `.bam` file was then used for downstream counting using homotrimer deduplication or UMI-tools. The resulting count tables were then used for differential gene expression analysis, which was performed using `DESeq2` v.1.40.2 (ref. 27) within the R statistical framework v.4.3.1.

UMI-tools deduplication

Following gene or transcript level mapping, the UMI was extracted from the read. Since UMI-tools was not designed to correct homotrimer sequences, we collapsed the UMI into a single nucleotide sequence by selecting the first base within each of the individual trimers. Reads were then deduplicated using the directional method using the command: `umi_tools count -per-gene -gene-tag=XT`.

Homotrimer set coverage deduplication

Following gene or transcript level mapping, the UMI was extracted from the read and collapsed into single nucleotide sequence using the majority vote approach where applicable or resolve inconsistencies through a combinatorial optimization scheme otherwise. Briefly, reads were first filtered to exclude reads in which there were more than three errors in the UMI sequence. For UMI sequences where each trimer contains at least two identical nucleotides, a majority vote was then performed to collapse the trimer into a monomer. If at least one trimer is inconclusive and contains three different nucleotides, we no longer treat each UMI sequence independently when collapsing trimers into monomers. Instead, we select one of the nucleotides in each trimer block to achieve maximal consistency between duplicates, that is to minimize the number of distinct collapsed UMI sequences. We formulate this task as a set cover problem for each gene as follows³¹. Let S be the set of sequenced homotrimer UMIs of a given gene (in a given cell). For $s \in S$ let $C(s)$ denote the set of collapsed UMIs that can be obtained by combining single nucleotides that occur in each trimer block of s . Each such collapsed sequence $c \in C(s)$, for some $s \in S$, can explain potentially multiple homotrimer UMIs s' if c is also contained in $C(s')$. We therefore include one subset $S_c \subseteq S$ for each $c \in \bigcup_{\{s \in S\}} C(s)$ that contains all $s \in S$ for which $c \in C(s)$. The collection of sets S_c of smallest cardinality that together include ('cover') all sequenced UMIs in S therefore corresponds to the smallest set of collapsed UMIs that explain all $s \in S$. To find this smallest set of collapsed UMIs, we use a greedy algorithm that starts from the empty set and in each iteration adds the subset S_c (that is, collapsed UMI c) that explains the largest number of yet unexplained sequenced UMIs. The solution returned by this algorithm is guaranteed to be within a logarithmic factor of the optimal solution³¹. In our experiments, the solution of the greedy approach was identical to the optimal solution for more than 90% of the genes. We computed the optimal solution using an integer linear programming approach, where decision variables model the inclusion or exclusion of sets S_c and linear inequalities enforce each sequenced UMI to be covered by at least one such set, that is to be explained by at least one collapsed UMI.

Settings for simulated UMIs

We simulated UMI data of length 30 (ten blocks of nucleotide trimers) to test the accuracy of our UMI correction methodology by using the `ResimPy` tool. We mimicked the PCR amplification and sequencing errors seen with ONT sequencing, as this sequencing methodology suffers from indels and base calling errors more frequently than PacBio or Illumina sequencing. UMIs were generated following an approach that was first described by UMI-tools¹¹. Briefly, we simulated homotrimer blocks of UMIs at random, with an amplification rate (`-ampl_rate`) ranging between 0.8 and 1.0 and then simulated PCR cycles so that each UMI was duplicated to the probability of amplification. PCR errors were then randomly added and assigned new probabilities of amplification. A predefined number of UMIs were randomly sampled to simulate sequencing depth and sequencing errors introduced with a specified probability. Finally, errors were detected by assessing the complementarity of homotrimers across the full UMI sequence. If no errors were detected, then the homotrimers were collapsed into single nucleotide bases. However, if errors were identified, then collapsing into single nucleotides was performed using the most common nucleotide within the trimer. If a most common nucleotide could not be determined, then a single nucleotide was selected at random for collapsing. The following values were used within our simulations. Sequencing depth 400; number of UMIs 50 (`-umi_num`); UMI length 12 (`-umi_len`); PCR error rate 3.6×10^{-6} (`-seq_err`); error rate $1 \times 10^{-1} - 1 \times 10^{-7}$ and number of PCR cycles 12 (`-pcr_num`); permutation tests 50 (`-perm_num`).

ResimPy: simulating PCR artifacts in UMI-attached reads

We developed `ResimPy` for simulating UMI-attached reads. The total number of reads $m^{(i+1)}$ at PCR cycle $i + 1$ comes from two sources: reads

that are PCR amplified and those that are not. This can be described by a Galton–Watson branching process^{32–34} as follows

$$m^{(i+1)} = m^{(i)} + n^{(i)} \quad (1)$$

Here, $n^{(i)}$ is the number of reads to be amplified, determined by an amplification rate α . According to Chen et al.³², $n^{(i)}$ follows a binomial distribution $\text{Binom}(m^{(i)}, \alpha)$. The $n^{(i)}$ reads to be amplified are randomly selected from the set $\{1, 2, \dots, m^{(i)}\}$ without replacement. This ensures that each read has an equal chance of being amplified. The same process applied for every two adjacent PCR cycles.

To simulate PCR errors, we implemented another Galton–Watson branching process. The total number of base errors $u^{(i+1)}$ at PCR cycle $i + 1$ is modeled by:

$$u^{(i+1)} = u^{(i)} + v^{(i)} \quad (2)$$

Here, $v^{(i)}$ represents the number of base errors to be synthesized at PCR cycle $i + 1$. Following Rabadan et al.³⁵, $v^{(i)}$ is generated using a negative binomial distribution $\text{NBinom}(r, q)$. Here q represents the probability of a base being successfully synthesized, which is derived by subtracting the base error probability P_e from 1 (that is $1 - P_e$). The variable r is determined by the product of the number of successfully synthesized bases calculated by $q \times t^{(i)}$ the positions of these $v^{(i)}$ PCR errors were randomly chosen from set $\{1, 2, \dots, t^{(i)}\}$, where $t^{(i)}$ represents the total number of bases to be synthesized at PCR cycle $i + 1$. Finally, the base at each error sequence position was substituted by one of the remaining three types of bases, drawn from a discrete uniform distribution $U(1, 3)$, where 1 and 3 represent the indices of the first and the third remaining bases, indicating that each one gains an equal chance for substitution. We use the same method to simulate sequencing errors. While simulation data provide some evidence for UMI deduplication performance, it is important to note that simulations can be biased. Therefore, we complement our simulations with experimentally derived data using our CMI approach described below.

CMI and error evaluation in bulk sequencing

To measure the error rate and evaluate the accuracy of our UMIs following library preparation and sequencing, we synthesized a common sequence (GGGAAACCCTTTGGGCCCTTAAACCCTTT) in replacement of a UMI to our polyA capture oligonucleotide. Following sequencing the CMI sequence was identified upstream of the SMART primer by pattern matching for GTACTCTGCGTTGATACCACTGCTT. The accuracy of our CMI was then determined by comparing the expected synthesized sequence to the extracted CMI sequence. The percentages of CMI that show full complementarity with the expected sequence were counted and the numbers of errors were determined for the inaccurate CMIs.

Comparison between UMI-tools and homotrimer CMI deduplication methods

After mapping the reads to the reference genome at the gene level, we processed the data using two different strategies: UMI-tools and homotrimer deduplication. For homotrimer deduplication, we used the full length of the CMI sequence, while for UMI-tools we collapsed the CMI into a monomer by selecting the first base for each trimer block. The inclusion of the CMI sequence to our reads provides an experimental ground truth with which to evaluate the accuracy of each deduplication strategy. To assess the accuracy of the final deduplicated counts, we compared them to the expected ground truth CMI gene count of 1.

10X Chromium library preparation

We prepared a single-cell suspension using JYN3 and 5TGM1 cells using the standard 10X Genomics chromium protocol as per the manufacturer's instructions. Briefly, cells were filtered into a single-cell suspension

using a 40 μM Flomi cell strainer before being counted. We performed 10X Chromium library preparation following the manufacturer's protocol. Briefly, we loaded 3,300 JYN3:5TGM1 cells at a 50/50 split into a single channel of the 10X Chromium instrument. Cells were barcoded and reverse transcribed into cDNA using the Chromium Single Cell 3' library kit and get bead v.3.1. We performed ten cycles of PCR amplification before cleaning up the library using 0.6 \times SPRI Select beads. The library was split and a further 20 or 25 PCR cycles were performed using a biotin oligonucleotide (5-PCBio-CTACACGACGCTCTCCGATCT) and then cDNA was enriched using Dynabeads MyOne streptavidin T1 magnetic beads (Invitrogen). The beads were washed in 2 \times binding buffer (10 mM Tris-HCl pH 7.5, 1 mM EDTA and 2 M NaCl), then samples were added to an equi-volume amount of 2 \times binding buffer and incubated at room temperature for 10 min. Beads were placed in a magnetic rack and then washed twice in 1 \times binding buffer. The beads were resuspended in H₂O and incubated at room temperature and subjected to long-wave ultraviolet light (~ 366 nm) for 10 min. Magnetic beads were removed, and library was quantified using the Qubit High-sensitivity kit. Libraries were then prepared before sequencing.

Drop-seq library preparation

Single-cell capture and reverse transcription were performed as previously described²⁰. Briefly, JYN3 and 5TGM1 cells (20:80 ratio) were filtered into a single-cell suspension using a 40 μM Flomi cell strainer before being counted. Cells were loaded into the DolomiteBio Nadia Innovate system at a concentration of 310 cells per μl . Custom synthesized beads were loaded into the microfluidic cartridge at a concentration of 620,000 beads per ml. Cell capture was then performed using the standard Nadia Innovate protocol according to the manufacturer's instructions. The droplet emulsion was then incubated for 10 min before being disrupted with 1H,1H,2H,2H-perfluoro-1-octanol (Sigma) and beads were released into aqueous solution. After several washes, the beads were subjected to reverse transcription. Before PCR amplification, beads were treated with ExoI exonuclease for 45 min. PCR amplification was then performed using the SMART PCR primer (AAGCAGTGGTATCAACGCAGAGT) and cDNA was subsequently purified using AMPure beads (Beckman Coulter). The library was split and a further 20 or 25 PCR cycles²⁰ were performed using a biotin oligonucleotide (5-PCBio-TACACGACGCTCTCCGATCT) and then cDNA was enriched using Dynabeads MyOne streptavidin T1 magnetic beads (Invitrogen). The beads were washed in 2 \times binding buffer (10 mM Tris-HCl pH 7.5, 1 mM EDTA and 2 M NaCl) then samples were added to an equi-volume amount of 2 \times binding buffer and incubated at room temperature for 10 mins. Beads were placed in a magnetic rack and then washed with twice with 1 \times binding buffer. The beads were resuspended in H₂O and incubated at room temperature and subjected to long-wave ultraviolet light (~ 366 nm) for 10 min. Magnetic beads were removed, and library was quantified using the Qubit High-sensitivity kit. Libraries were then prepared for sequencing.

Bulk and single-cell library preparation and ONT sequencing

A total of 500 ng of single-cell PCR input was used as a template for ONT library preparation. Library preparation was performed using the SQK-LSK114 (kit V14) ligation sequencing kit, following the manufacturer's protocol. Samples were then sequenced on either a Flongle device or a PromethION device using R10.4 (FLO-PRO114M) flow cells.

10X analysis workflow

We performed base calling on the raw fast5 data to generate fastq files using Guppy (v.6.4.8) (guppy_basecaller -compress-fastq-cdna_r10.4_e8.1_sup.cfg-x'cuda:0') in GPU mode from ONT running on a RTX3090 graphics card. To process the 10X chromium data, we wrote a custom cgatcore pipeline (https://github.com/cribbslab/TallyTriN/blob/main/tallytrin/pipeline_10x.py)²². We first determined the orientation

of the reads and if a poly-T sequence was detected we reverse complemented the read. Next, we identified the barcode and UMI based on the pairwise alignment of the sequence AGATCGGAAGAGCGT and AAAAAAAAA and identified the sequence between these alignments. We next removed reads that were greater or equal to 28 bp and isolated the barcode as the first 16 bp and the UMI the following 12 bp. The barcode and UMI sequence were then appended to the name of the fastq read using the underscore delimiter. Next, to remove barcode errors we parsed the barcodes from each read in the fastq file and then selected the most common barcode sequences using the number of expected cells in our library as the threshold. Next, for every read in the fastq file we then identified the closest barcode match for each read, allowing for two mismatches. Mapping was performed using minimap2 (v.2.25)³⁶, with the following settings: `-ax splice -uf -MD -sam-hit-only -junc-bed` and using the reference transcriptome for human hg38 and mouse mm10. The resulting .bam file was sorted and indexed before adding the transcript name to the XT tag within the .bam file. Counting was then performed using UMI-tools `-method=directional` before being converted to a market matrix format. Raw transcript expression matrices generated by UMI-tools count were processed using R/Bioconductor (v.4.3.0), the raw market matrix files were imported into R using bustools (v.0.42.0) and the Seurat^{37,38} package (v.4.3.0). Transcript matrices were cell-level scaled and log-transformed. The top 2,000 highly variable genes were then selected based on variance stabilizing transformation that was used for principal component analysis. Clustering was performed within Seurat using the Louvain algorithm. To visualize the single-cell data, we projected data onto a uniform manifold approximation and projection (UMAP)³⁹.

Drop-seq analysis workflow

We performed base calling on the raw fast5 data to generate fastq files using Guppy (v.6.4.8) (`guppy_basecaller -compress-fastq -c dna_r10.4_e8.1_sup.cfg -x 'cuda:0'`) in GPU mode from ONT running on a RTX3090 graphics card. To process the drop-seq data, we wrote a custom cgatcore pipeline (<https://github.com/cribbslab/TallyTriN>)²². We followed the workflow previously described for identifying barcodes and UMIs using scCOLOR-seq sequencing analysis¹³. Briefly, to determine the orientation of our reads, we first searched for the presence of a polyA sequence or a poly-T sequence. In cases where the poly-T was identified, we reverse complemented the read. We next identified the barcode sequence by searching for the polyA region and flanking regions before and after the barcode. The trimer UMI was identified based on the primer sequence GTACTCTGCGTT at the TSO distal end of the read, allowing for two mismatches. Barcodes and UMIs that had a length less than 48 base pairs were filtered. To conduct monomer-based analyses, a random base was selected from each homotrimer in the UMI or CMI and collapsed into a monomer. Homotrimer UMI correction was performed following mapping using minimap2 (v.2.25)³⁶. Mapping settings were as follows: `-ax splice -uf -MD -sam-hit-only -junc-bed` and using the reference transcriptome for human hg38 and mouse mm10. The resulting .sam file was sorted and indexed using samtools²⁵. For monomer UMI, counting was performed using UMI-tools before being converted to a market matrix format. For homotrimer UMI correction, the counting was performed using the script `greedy.py` within the TallyTriN repository. Raw transcript expression matrices generated by UMI-tools count and `greedy.py` were processed using R and Bioconductor (v.4.3.0) and custom scripts were used to generate barnyard plots showing the proportion of mouse and human cells. Transcript matrices were cell-level scaled and center log ratio transformed. The top 3,000 highly variable genes were then selected based on variance stabilizing transformation that was used for principal component analysis. Clustering was performed within Seurat using the Louvain algorithm. To visualize the single-cell data, we projected data onto a UMAP³⁹.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

Sequencing data have been deposited to the Gene Expression Omnibus under the accession number [GSE218899](https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE218899). All analysis was performed using hg38 ensembl 98 version.

Code availability

Source data are provided with this paper. All custom pipelines used within this analysis are available on GitHub (<https://github.com/cribbslab/TallyTriN>). ResimPy (v.0.0.1) is available on GitHub (<https://github.com/cribbslab/resimpy>).

References

- Zhu, Y. Y., Machleder, E. M., Chenchik, A., Li, R. & Siebert, P. D. Reverse transcriptase template switching: a SMART approach for full-length cDNA library construction. *Biotechniques* **30**, 892–897 (2001).
- Cribbs, A. et al. CGAT-core: a python framework for building scalable, reproducible computational biology workflows [version 1; peer review: 1 approved, 1 approved with reservations]. *F1000 Res.* **8**, 377 (2019).
- FastQC: a quality control tool for high throughput sequence data (Brabham Bioinformatics, 2010).
- Ewels, P., Magnusson, M., Lundin, S. & Kaller, M. MultiQC: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics* **32**, 3047–3048 (2016).
- Li, H. et al. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
- Liao, Y., Smyth, G. K. & Shi, W. featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* **30**, 923–930 (2014).
- Love, M. I., Huber, W. & Anders, S. Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol.* **15**, 550 (2014).
- Cribbs, A. et al. CGAT-core: a python framework for building scalable, reproducible computational biology workflows [version 2; peer review: 1 approved, 1 approved with reservations]. *F1000 Res.* **8**, 377 (2019).
- Kim, D., Paggi, J. M., Park, C., Bennett, C. & Salzberg, S. L. Graph-based genome alignment and genotyping with HISAT2 and HISAT-genotype. *Nat. Biotechnol.* **37**, 907–915 (2019).
- Liao, Y., Smyth, G. K. & Shi, W. The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote. *Nucleic Acids Res.* **41**, e108 (2013).
- Chvatal, V. A greedy heuristic for the set-covering problem. *Math. Oper. Res.* **4**, 233–235 (1979).
- Chen, Y. J. et al. Quantifying molecular bias in DNA data storage. *Nat. Commun.* **11**, 3264 (2020).
- Lalam, N. Statistical inference for quantitative polymerase chain reaction using a hidden Markov model: a Bayesian approach. *Stat. Appl. Genet. Mol. Biol.* **6**, 10 (2007).
- Wagner, A. et al. Surveys of gene families using polymerase chain-reaction—PCR selection and PCR drift. *Syst. Biol.* **43**, 250–261 (1994).
- Rabadan, R. et al. On statistical modeling of sequencing noise in high depth data to assess tumor evolution. *J. Stat. Phys.* **172**, 143–155 (2018).
- Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
- Stuart, T. et al. Comprehensive integration of single-cell data. *Cell* **177**, 1888–1902 e1821 (2019).

38. Hao, Y. et al. Integrated analysis of multimodal single-cell data. *Cell* **184**, 3573–3587 e3529 (2021).
39. McInnes, L., Healy, J., Saul, N. & Großberger, L. UMAP: uniform manifold approximation and projection. *J. Open Source Softw.* **3**, 861 (2018).

Acknowledgements

Research support was obtained from Innovate UK (T.B. Sr, T.B. Jr, U.O., M.P., A.P.C.), the Engineering and Physical Sciences Research Council (U.O., M.P., A.P.C.), the National Institute for Health Research Oxford Biomedical Research Unit (U.O.), Cancer Research UK (U.O. and A.P.C.), the Bone Cancer Research Trust (A.P.C. and U.O.), the Leducq Epigenetics of Atherosclerosis Network program grant from the Leducq Foundation (U.O.), the Chan Zuckerberg Initiative (A.P.C.) and the Myeloma Single Cell Consortium (U.O.). A.P.C. is a recipient of a Medical Research Council career development fellowship (grant no. MR/V010182/1). S.L. holds a fellowship from the China Scholarship Council (grant no. CSC202006230065) and P.M.M. holds a Deutsche Forschungsgemeinschaft fellowship through the Graduate School of Quantitative Biosciences Munich. We thank C. Edwards, N. Athanasou and B. Hassan for sharing access to cell lines.

Author contributions

A.P.C. designed the study with contributions from M.P., T.B. Jr, T.B. Sr, S.C. and U.O. A.P.C. also conducted data analysis, generated the figures and wrote the paper with input from all authors. J.S. conducted

data analysis, generated figures and wrote the paper. M.P., D.L., S.H., G.H., J.R., N.M. and V.G. performed experiments. S.L. and P.M.-M. undertook set combinatorial method development and computational analyses.

Competing interests

A.P.C., U.O. and M.P. are cofounders of Caeruleus Genomics Ltd and are inventors on several patents related to sequencing technologies filed by Oxford University Innovations. T.B. Jr is a director of ATDBio. T.B. Sr is a consultant of ATDBio. The other authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s41592-024-02168-y>.

Correspondence and requests for materials should be addressed to Adam P. Cribbs.

Peer review information *Nature Methods* thanks Daniel Hoffmann, Ian Sudbery and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Lei Tang and Hui Hua, in collaboration with the *Nature Methods*.

Reprints and permissions information is available at www.nature.com/reprints.

Corresponding author(s): Adam P Cribbs

Last updated by author(s): Nov 24, 2023

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our [Editorial Policies](#) and the [Editorial Policy Checklist](#).

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a Confirmed

- The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided
Only common tests should be described solely by name; describe more complex techniques in the Methods section.
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g. F , t , r) with confidence intervals, effect sizes, degrees of freedom and P value noted
Give P values as exact values whenever suitable.
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen's d , Pearson's r), indicating how they were calculated

Our web collection on [statistics for biologists](#) contains articles on many of the points above.

Software and code

Policy information about [availability of computer code](#)

Data collection

For PacBio CCS fastq reads were generated using CCS v6.3.0 (<https://github.com/PacificBiosciences/ccs>). For ONT sequencing, fastq files were generated using guppy v6.4.8 (ONT downloads)

Data analysis

Custom pipelines used within the analysis is available on Github (<https://github.com/cribbslab/TallyTriN>). External software called by these pipelines include: minimap2 (v2.25); Seurat package (v4.3.0); R/Bioconductor (v4.3.0); pysam (v0.21.0) bustools (v0.42.0); cgatcore (v0.6.15); UMI-tools (v1.1.4); featureCounts (v2.0.3) and hisat2 (v2.2.1). ResimPy (v0.0.1) is available on github (<https://github.com/cribbslab/resimpy>).

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio [guidelines for submitting code & software](#) for further information.

Data

Policy information about [availability of data](#)

All manuscripts must include a [data availability statement](#). This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our [policy](#)

Sequencing data has been deposited to GEO under accession number GSE218899. All analysis was performed using hg38 ensembl 98 version.

Human research participants

Policy information about [studies involving human research participants and Sex and Gender in Research](#).

Reporting on sex and gender	<input type="text" value="N/A"/>
Population characteristics	<input type="text" value="N/A"/>
Recruitment	<input type="text" value="N/A"/>
Ethics oversight	<input type="text" value="N/A"/>

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

- Life sciences Behavioural & social sciences Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see [nature.com/documents/nr-reporting-summary-flat.pdf](https://www.nature.com/documents/nr-reporting-summary-flat.pdf)

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	<p>No sample size calculation was performed prior to the study. We chose a minimum of three independent experiments to evaluate our methodology, a decision guided by several key considerations.</p> <p>Firstly, the principle of replication, which is fundamental to scientific robustness, informed our choice. By conducting the experiment three times independently, we aimed to ensure that our findings were reproducible and not simply due to random variation.</p> <p>Secondly, our methodology's primary focus is on evaluating the robustness of our assay. Utilising three independent experiments strikes a balance between practical feasibility and obtaining sufficient data to assess the assay's consistency across different runs. This number of experiments was deemed to provide enough data points to enable a meaningful analysis while being feasible in terms of resource allocation and time management.</p> <p>Additionally, with a sample size of three, we are able to calculate the standard error of the mean (SEM). The SEM is an essential statistical measure for our study, particularly for the UMI/CMI correction, as it provides an estimate of the precision of our sample mean. This is crucial in quantifying the uncertainty associated with the measurements obtained from our assay.</p>
Data exclusions	<input type="text" value="No data was excluded from the study."/>
Replication	<input type="text" value="All of our experiments were replicated in a minimum of three independent experiments unless otherwise stated within the figure legends."/>
Randomization	<p>A comparative analysis was not undertaken, and therefore blinding was not deemed necessary. However, to provide clarity on the experimental design, we will elaborate on how samples were allocated into experimental groups.</p> <p>In this study, sample allocation into experimental groups was not performed randomly. This decision was based on the specific objectives and design of our research, which focused on the evaluation of a particular assay or process rather than comparing different groups or conditions. Given the non-comparative nature of the study, the allocation of samples was guided primarily by the need to test the assay under consistent and controlled conditions.</p> <p>We ensured that the samples were representative of the typical conditions under which the assay would be used, but they were not distributed across different experimental groups for comparative analysis. Therefore, the control of covariates, a critical aspect in comparative</p>

studies to ensure that differences between groups are not due to confounding variables, was not applicable in the context of our study.

Blinding

Blinding was not performed as part of this study. In the our study, the primary aim was to evaluate the performance of a specific assay under consistent conditions, rather than to compare outcomes across different groups or conditions. The focus was on the objective measurement of assay parameters, which are inherently independent of researcher or participant bias. Given the nature of the measurements—likely quantitative and based on predefined, objective criteria—the potential for subjective interpretation or bias was minimal.

Furthermore, the experimental design did not involve variables that could be influenced by the knowledge of the researchers or subjects. For instance, if the study involved chemical or laboratory analyses where outcomes are determined by standardized procedures and equipment, the scope for bias that could be mitigated by blinding is significantly reduced.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involvement in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> Antibodies
<input type="checkbox"/>	<input checked="" type="checkbox"/> Eukaryotic cell lines
<input checked="" type="checkbox"/>	<input type="checkbox"/> Palaeontology and archaeology
<input checked="" type="checkbox"/>	<input type="checkbox"/> Animals and other organisms
<input checked="" type="checkbox"/>	<input type="checkbox"/> Clinical data
<input checked="" type="checkbox"/>	<input type="checkbox"/> Dual use research of concern

Methods

n/a	Involvement in the study
<input checked="" type="checkbox"/>	<input type="checkbox"/> ChIP-seq
<input checked="" type="checkbox"/>	<input type="checkbox"/> Flow cytometry
<input checked="" type="checkbox"/>	<input type="checkbox"/> MRI-based neuroimaging

Eukaryotic cell lines

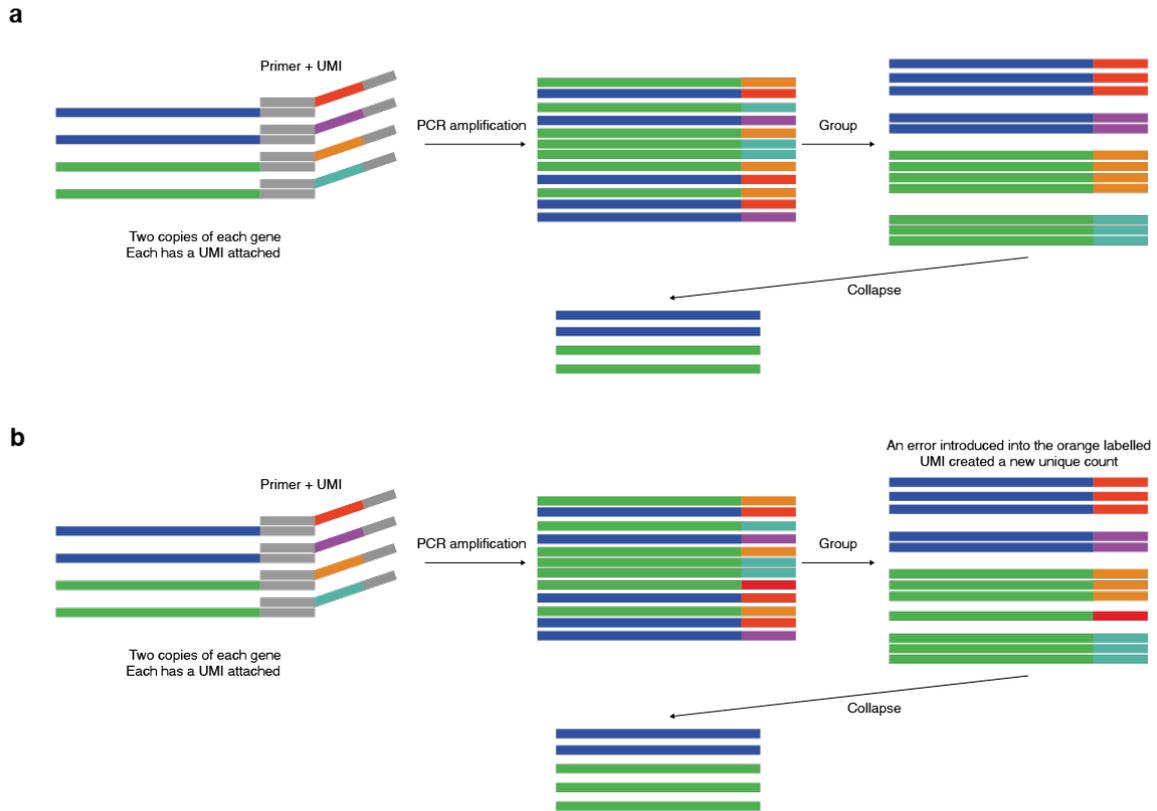
Policy information about [cell lines and Sex and Gender in Research](#)

Cell line source(s)	Jurkat and RM82 cell lines were purchased from ATCC. STGM1 cells were a kind gift from Prof Clair Edwards (University of Oxford).
Authentication	Cell lines were authenticated by STR.
Mycoplasma contamination	Cell lines were mycoplasma tested routinely and all tested negative.
Commonly misidentified lines (See ICLAC register)	No commonly misidentified lines used in this study.



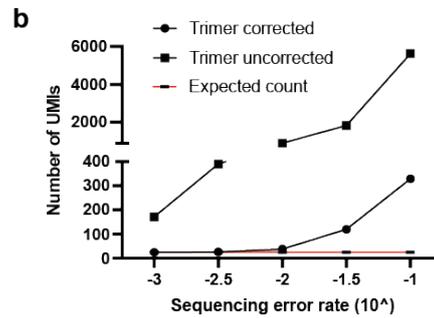
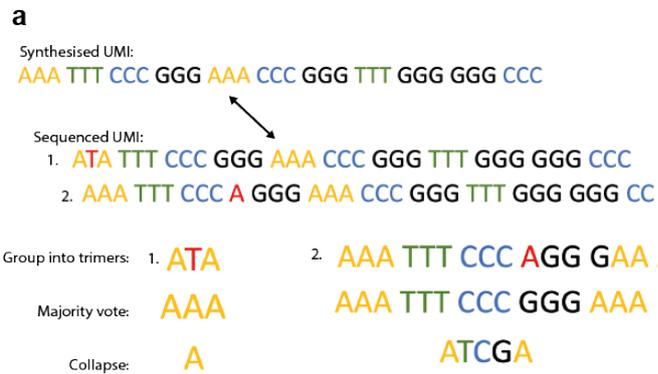
Correcting PCR amplification errors in unique molecular identifiers to generate accurate numbers of sequencing molecules

In the format provided by the authors and unedited



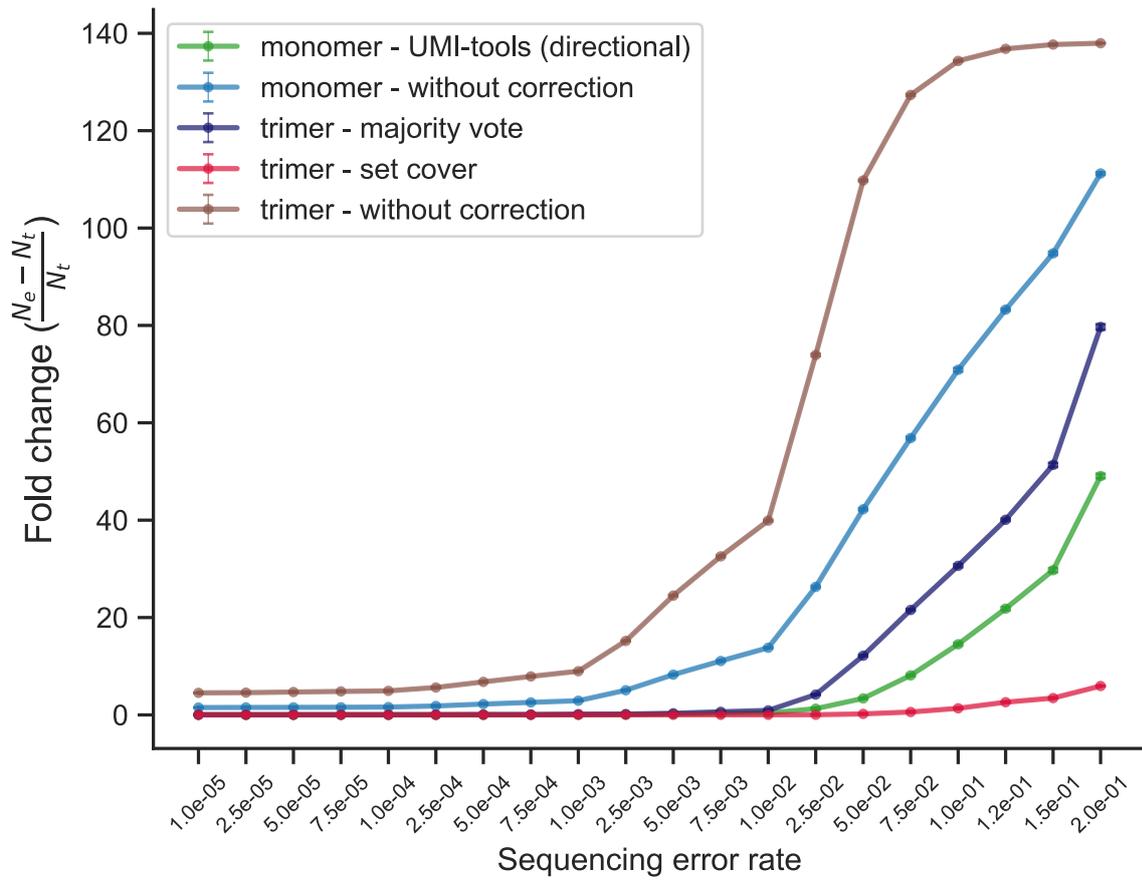
Supplementary Figure 1: UMI errors from sequencing or PCR increase UMI counting.

a, Ideal UMI collapsing example: This figure illustrates an ideal scenario where transcripts (two blue and two green) are labelled with unique molecular identifier barcodes (UMIs) and PCR amplification is performed. Due to PCR amplification bias, longer transcripts have a lower amplification rate than shorter ones. The sequenced reads are then grouped together based on the set of UMIs and then collapsed within those groups to match the original number of transcripts. **b**, Increased UMI counting occurs due to errors: In real situations, errors occur during PCR amplification and sequencing, which can lead to increased UMI counts. As shown in this example, an error occurs within one of the UMIs which results in a higher count of unique UMIs than the actual number of input transcripts in the final library. This phenomenon can affect downstream analysis and should be taken into consideration otherwise they will lead to false positives when performing differential expression analysis.



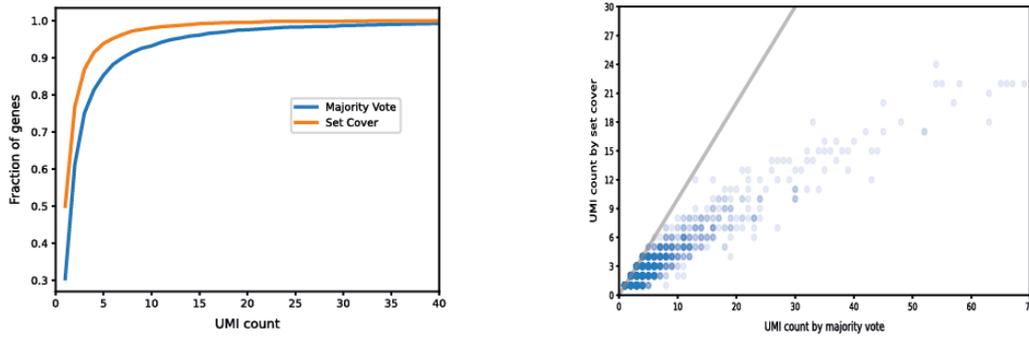
Supplementary Figure 2: Improved UMI deduplication using homotrimer blocks of nucleotides.

a, Homotrimer blocks of nucleotides are used to synthesise UMIs, which enables efficient error correction through a majority vote between the trimer blocks. This approach does not require knowledge of the original oligonucleotide, and the trimers are resilient to base pair errors (example 1) and insertions/deletions (example 2). Errors are removed before collapsing the sequence to a single-base and then performing downstream analyses. **b**, We simulated UMIs with increasing error rates were modelled the correction of trimer sequences using the majority vote approach as described in **a**. To handle homotrimer errors more robustly, we subsequently developed a new model, which is described within the methods section “homotrimer correction”. Error bars in **b** are plotted but not visible because the CVs are very small over three different repeated simulations. Error bars represent the mean and +/- Standard Deviation.



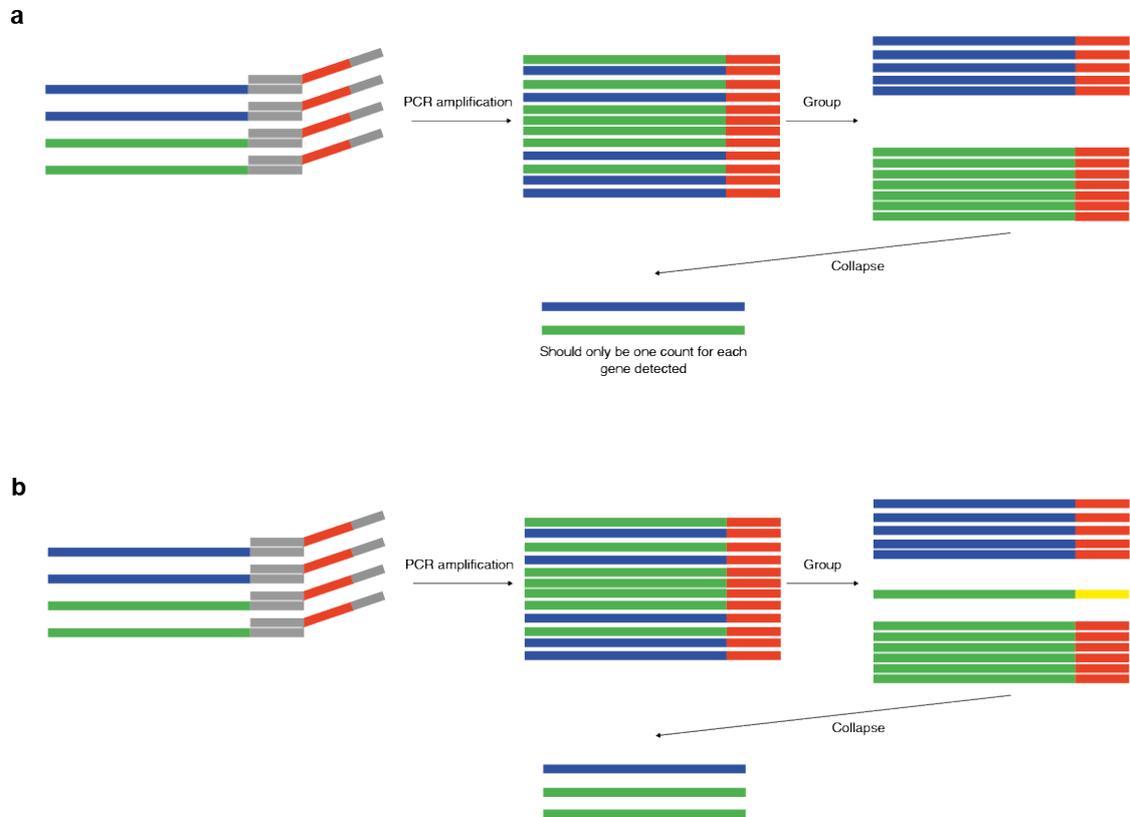
Supplementary Figure 3: Comparison of homotrimer and monomer UMI for demultiplexing.

We evaluated the effectiveness of homotrimer UMIs versus monomer UMIs for demultiplexing by simulating base UMIs with increasing error rates and calculated the fold change between the ground truth and the output after computational error correction is applied. Our findings demonstrate that the performance of correcting homotrimer UMIs using the set coverage approach outperform that of uncorrected monomer UMIs and correcting monomer UMIs using computational methods. PCR cycles were set to 12. Sequencing depth was set to 400. The number of initial molecules was set to 50. The length of homotrimer UMIs was set to 36bp. N_e represents the number of sequenced molecules. N_t represents the number of initial molecules. Error bars are s.d. of 3 independent simulations.



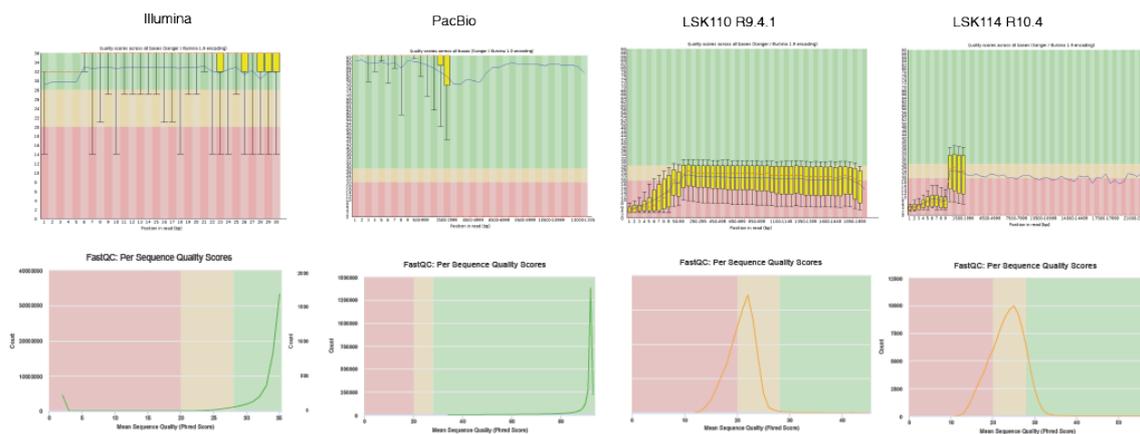
Supplementary Figure 4: The majority vote method is improved using a set coverage solution.

a, A cumulative plot showing the fraction of genes that have been collapsed to a common molecular identifier (CMI) count of less than or equal to the value shown on the x-axis by either the majority vote approach or the set cover-based optimization method. A CMI refers to a common homotrimer sequence attached to every captured RNA molecule, when demultiplexed without errors, the RNA count will equal 1, any errors will inflate the RNA molecule count. The data used for this approach is ONT sequencing using R9.4 and LSK110 chemistry. Only genes with at least 2 mapped reads were considered ($n=3,428$). Maximal UMI counts returned by majority vote and set cover optimization are 245 and 72, respectively. **b**, A scatter plot comparing UMI counts obtained using the majority vote approach (x-axis) to counts returned by the greedy set cover algorithm (y-axis). Only genes with at least 2 mapped reads were considered ($n=3,428$). To simplify visualization, we excluded genes with large majority counts $(x,y) = (83, 22), (91, 24), (96, 31), (106, 32), (113, 33)$ and $(245, 72)$.



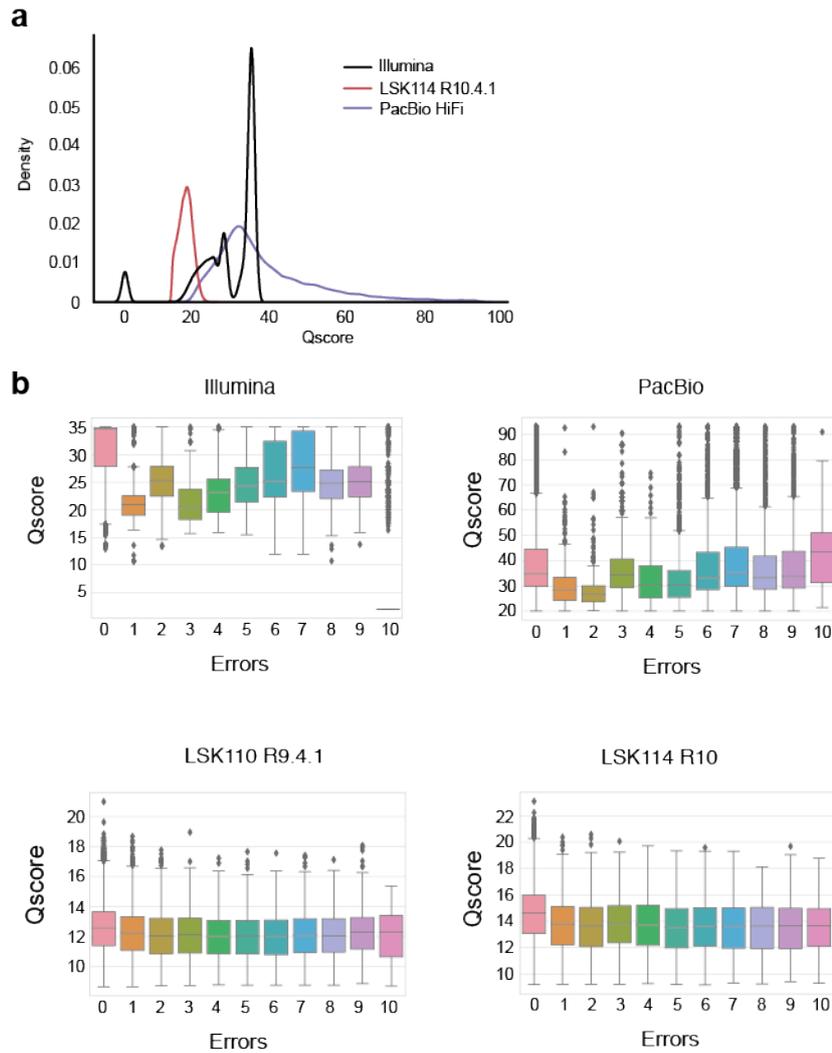
Supplementary Figure 5: Empirical evaluation of transcript counting with Common Molecular Identifiers (CMIs).

a, An Ideal CMI collapsing example: In this scenario, transcripts (two blue and two green) are labelled with a common molecular identifier barcode (CMIs; labelled as red) and amplified via PCR. During transcripts grouping, all transcripts are labelled with the same common sequence. Therefore, following demultiplexing, each transcript should receive a count of one for every instance of detection. **b**, Increased counts result from the introduction of errors: This figure illustrates the effect of the errors within the CMI sequence. Any error introduced during PCR or sequencing creates a new CMI (labelled as yellow), resulting in an increase in transcript counts. This allows empirical evaluation of the effect of errors on the counting of transcripts, providing valuable insights into the accuracy of transcript quantification.



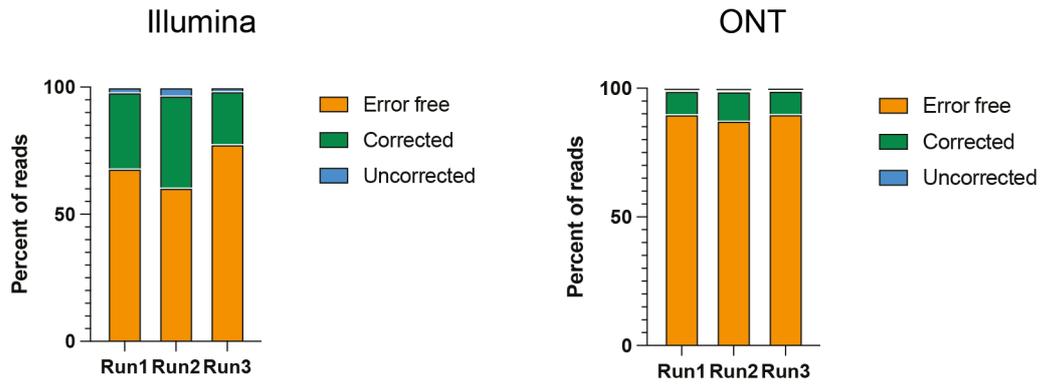
Supplementary Figure 6: Read quality control following sequencing by Illumina, PacBio and ONT.

The read quality outputs from FASTQC for Illumina, PacBio and ONT (old chemistry: LSK110 R9.4.1 and new kit14 chemistry LSK114 R10.4). Error bars are defined by FASTQC and the central red line is the median value, the yellow box represents the inter-quartile range (25-75%), the upper and lower whiskers represent the 10% and 90% points, the blue line represents the mean quality.



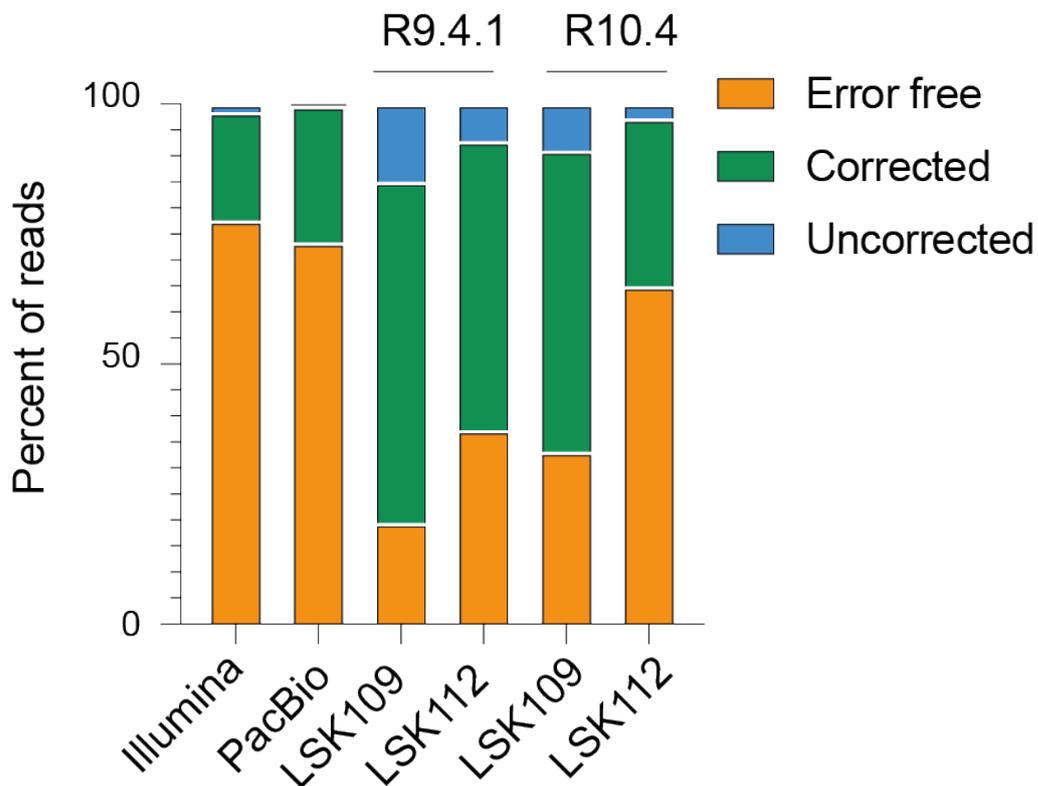
Supplementary Figure 7: Quality score following sequencing by Illumina, PacBio and ONT.

The Qscore of each read as a relationship between the number of errors measured within the CMI, sequencing across Illumina, PacBio and ONT (old chemistry: LSK110 R9.4.1 and new kit14 chemistry LSK114 R10.4) technologies. **a**, Qscore represented as a density plot for the different sequencing platforms. **b**, The Qscore relationship with the number of errors detected within the CMI across the different sequencing platforms. The Error bars are plotted so that the central line is the median value, the box represents the inter-quartile range (25-75%), the upper and lower whiskers represent the 10% and 90% points, the points represent the extreme outliers.



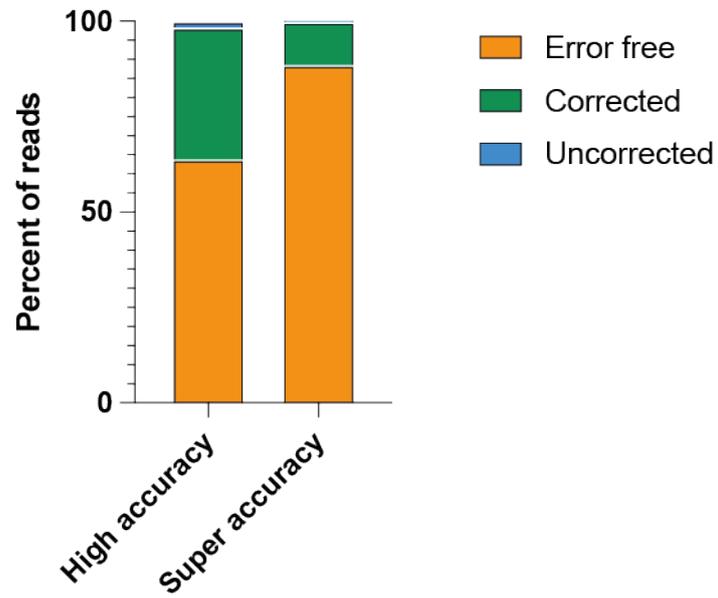
Supplementary Figure 8: Percent of correctly sequenced CMIs across Illumina and ONT sequencing platforms

The left panel shows the individual repeats that were performed following Illumina sequencing of a homotrimeric CMI tagged cDNA. Each run was performed independently and sequenced using separate flow cells. The right panel shows the same cDNA sequenced using the ONT platform, across three separate minION flow cells.



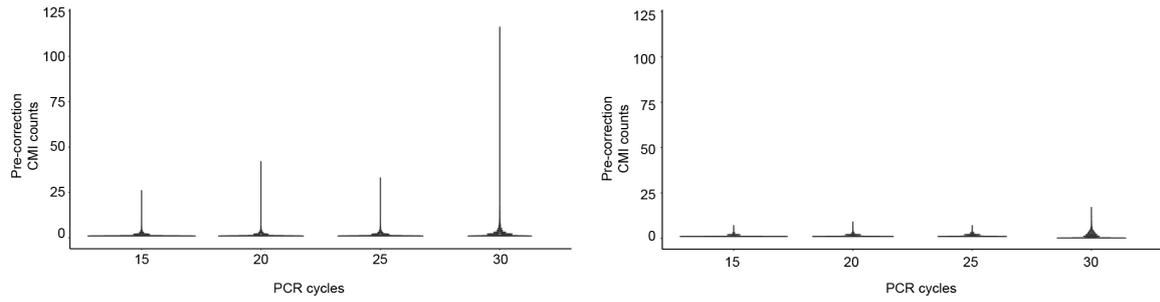
Supplementary Figure 9: Evaluation of CMI accuracy across Illumina, PacBio and legacy ONT chemistry.

The accuracy of sequencing was evaluated for legacy ONT chemistry, we measured the percentage of CMIs with a Hamming distance between the expected and the sequenced CMI. The results are shown for Illumina, PacBio, and ONT legacy chemistry sequencing. Data from Illumina and PacBio is the same as in Fig. 1h. Data from Illumina and ONT were performed in triplicate, whereas PacBio was performed as a single run.



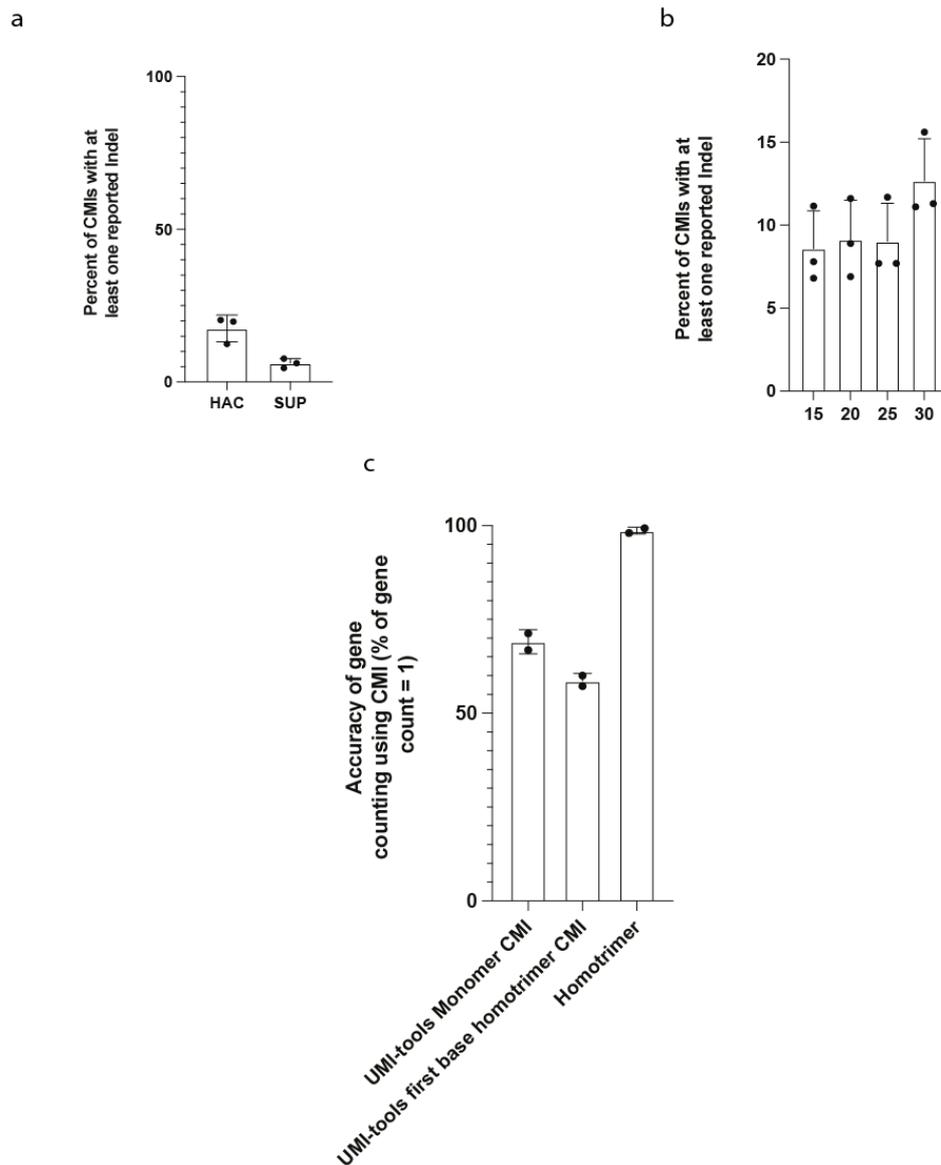
Supplementary Figure 10: Improved basecalling accuracy using super accuracy guppy basecalling for ONT technology.

Percent of CMIs that are correctly sequenced and then error corrected using homotrimer correction using either high accuracy guppy basecalling or super accuracy basecalling for the LSK114 chemistry and R10.4 flow cells.



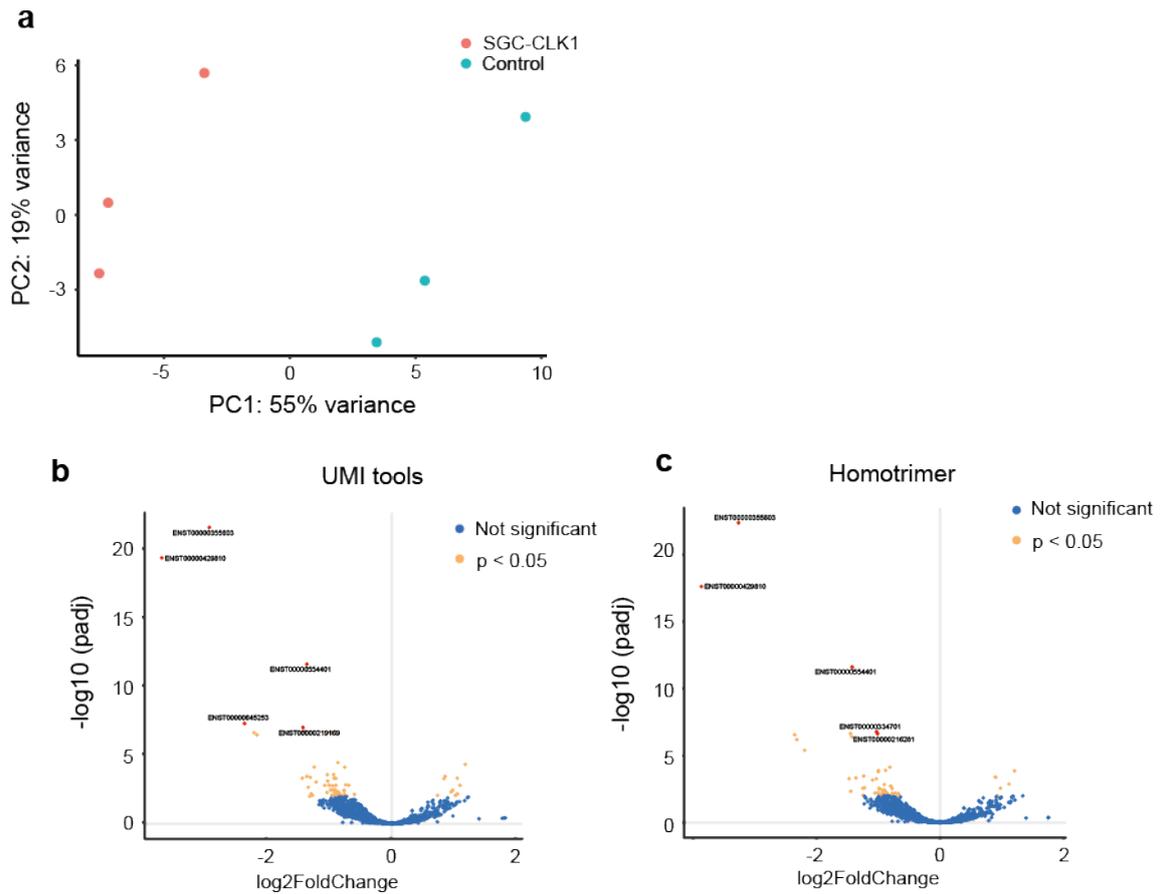
Supplementary Figure 11: CMI counts pre and post majority vote corrected.

The left panel shows a violin plot of the counts for each transcript pre majority vote correction following 15, 20, 25 and 30 PCR cycles. The right-hand panel shows the CMI counts post majority vote correction. The ground truth count for each transcript should be equal to 1, any counts above this indicate an error.



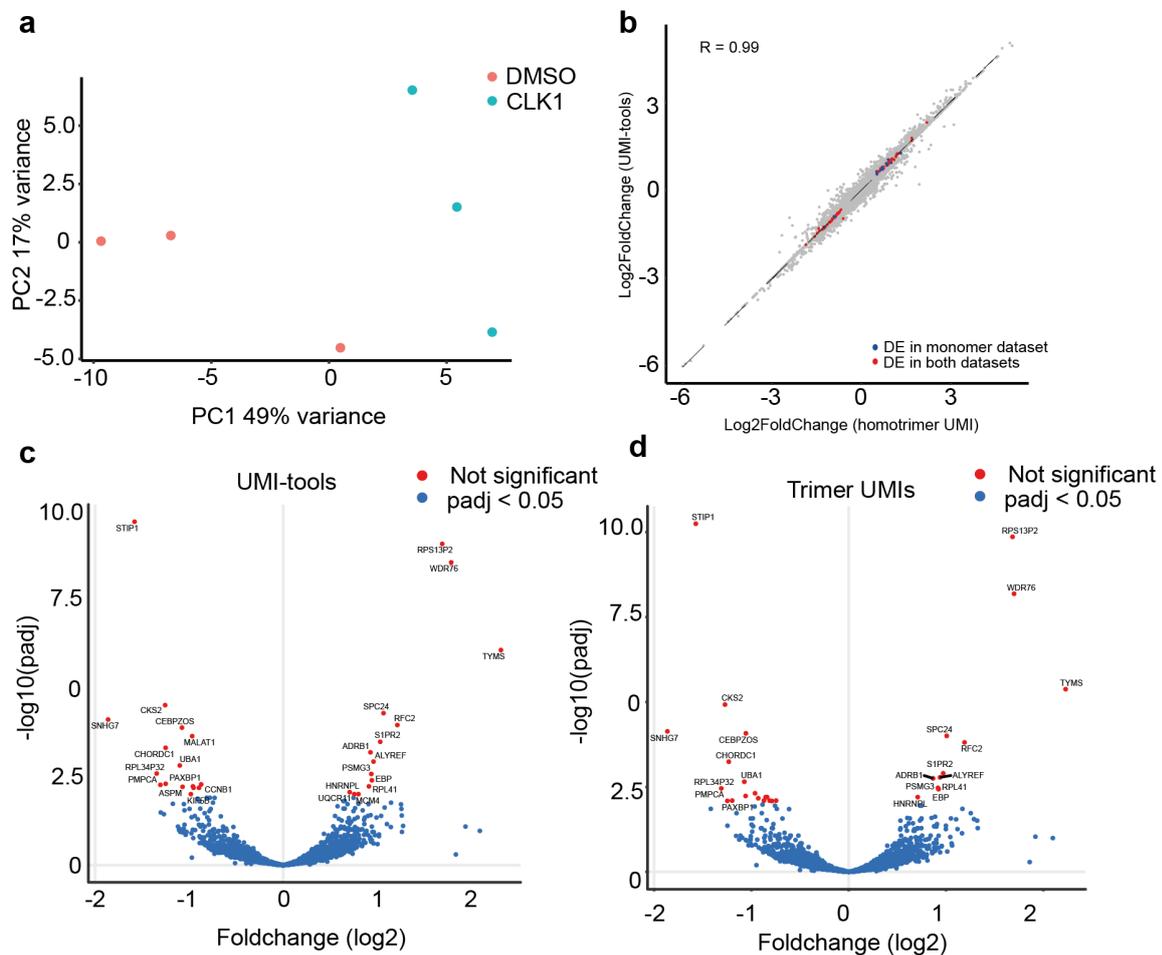
Supplementary Figure 12: The indel frequency and accuracy of counting using different CMI approaches.

a, The percent of CMIs with at least one indel were calculated for a bulk homotrimer CMI sequencing experiment using both High accuracy (HAC) and Super accuracy basecalling (SUP). **b**, The percent of CMIs with at least one indel were calculated for a bulk homotrimer CMI sequencing following 15, 20, 25 and 30 PCR cycles. **c**, This figure represents the percentage of genes with accurately quantified CMI after applying ONT sequencing. Three strategies were compared: (1) counting using a monomer-synthesised CMI and subsequent application of UMI-tools, (2) employing a homotrimer-synthesised CMI with a selection of the first base in the trimer block and (3) using the homotrimer error correction. Error bars are s.d. of 3 independent experiments.



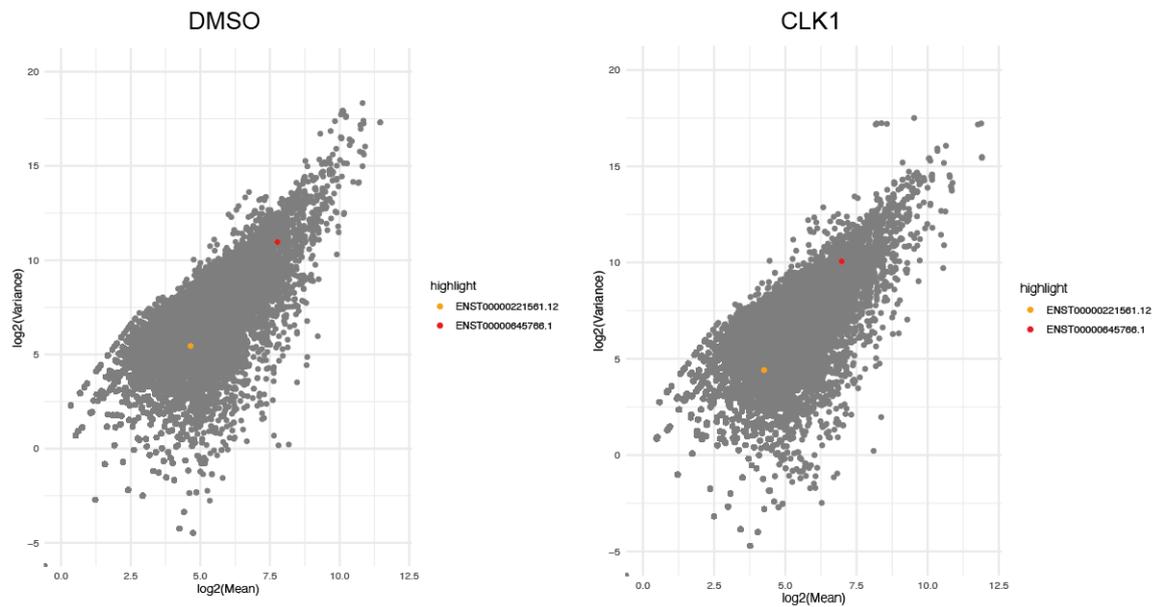
Supplementary Figure 13: Analysis of differential gene expression in RM82 Ewing's sarcoma cells treated with DMSO and CLK-1 inhibitor and sequenced using the ONT platform.

a, A PCA plot showing the variance for cells treated with either DMSO or CLK-1 inhibitor. **b**, A volcano plot showing the log₂ fold change and -log₁₀ padj values for cells treated with DMSO or CLK-1 inhibitor, analysed without the inclusion of a UMI during analysis. **d**, A volcano plot showing the log₂ fold change and -log₁₀ padj values for cells treated with DMSO or CLK-1 inhibitor and analysed using the homotrimer corrected UMI. These results demonstrate the utility of homotrimer correction in identifying differentially expressed genes and removal of false positive transcripts. Differential expression was determined using DESeq2 using a Wald test and a p adjusted value of < 0.05 was used as a threshold.



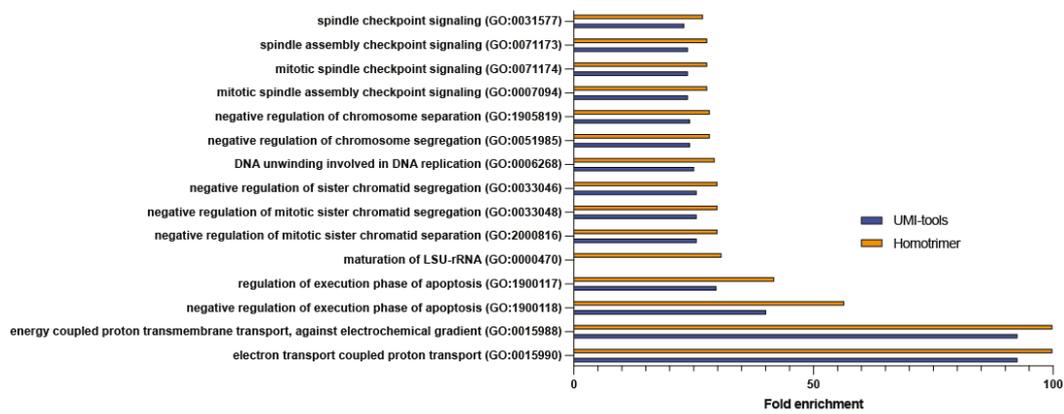
Supplementary Figure 14: Analysis of differential gene expression in RM82 Ewing's sarcoma cells treated with DMSO and CLK-1 inhibitor and sequenced using the Illumina platform.

a, A PCA plot showing the variance for cells treated with either DMSO or CLK-1 inhibitor. **b**, This scatter plot compares the log₂ fold changes obtained from randomly collapsing each sequenced trimer UMI with those obtained from homotrimer UMI correction. **c**, A volcano plot showing the log₂ fold change and -log₁₀ padj values for cells treated with DMSO or CLK-1 inhibitor, analysed without the inclusion of a UMI during analysis. **d**, A volcano plot showing the log₂ fold change and -log₁₀ padj values for cells treated with DMSO or CLK-1 inhibitor and analysed using the homotrimer corrected UMI. These results demonstrate the utility of homotrimer correction in identifying differentially expressed genes and removal of false positive genes. Differential expression was determined using DESeq2 using a Wald test and a p adjusted value of < 0.05 was used as a threshold.



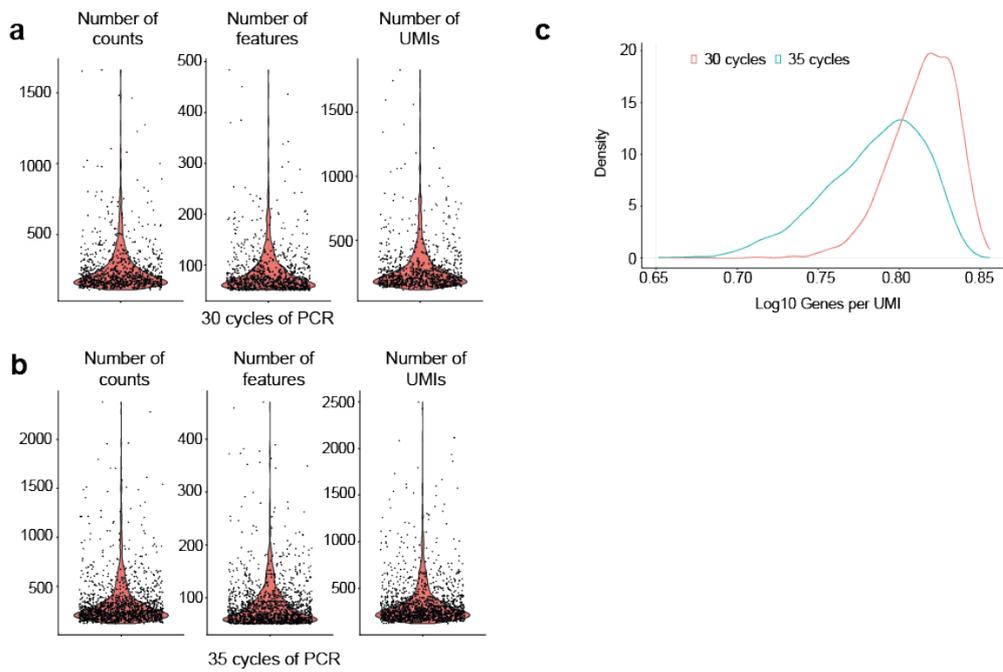
Supplementary Figure 15: Mean variance plot of transcripts showing samples treated with DMSO and CLK1.

The mean variance plots show that the transcript expression is correlated, with the higher the transcript expression the higher the variance. The data follows a negative binomial distribution. Two transcripts plotted in Fig. 1m-n are plotted as red and orange points.



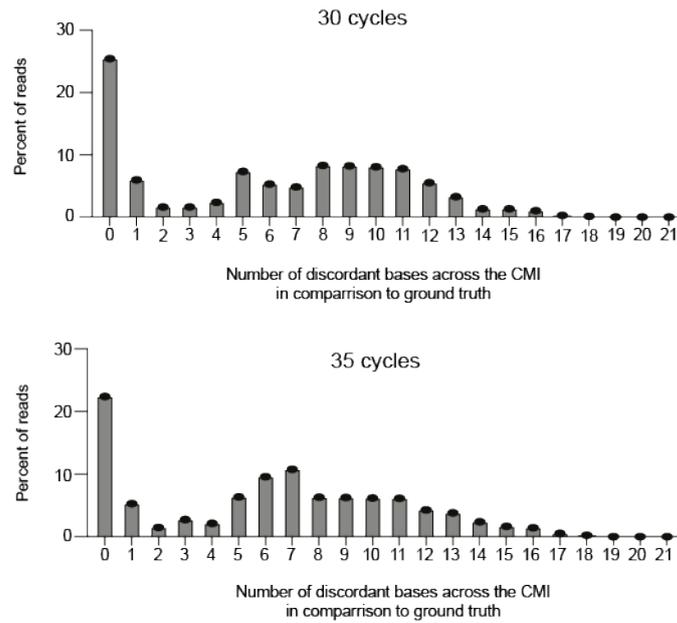
Supplementary Figure 16: GO analysis of the differentially expressed genes between DMSO and CLK-1 inhibitor.

Go analysis was performed for data shown in Fig. 1I that corrected using UMI-tools (blue bars) and differentially regulated genes following homotrimer correction (orange bars).



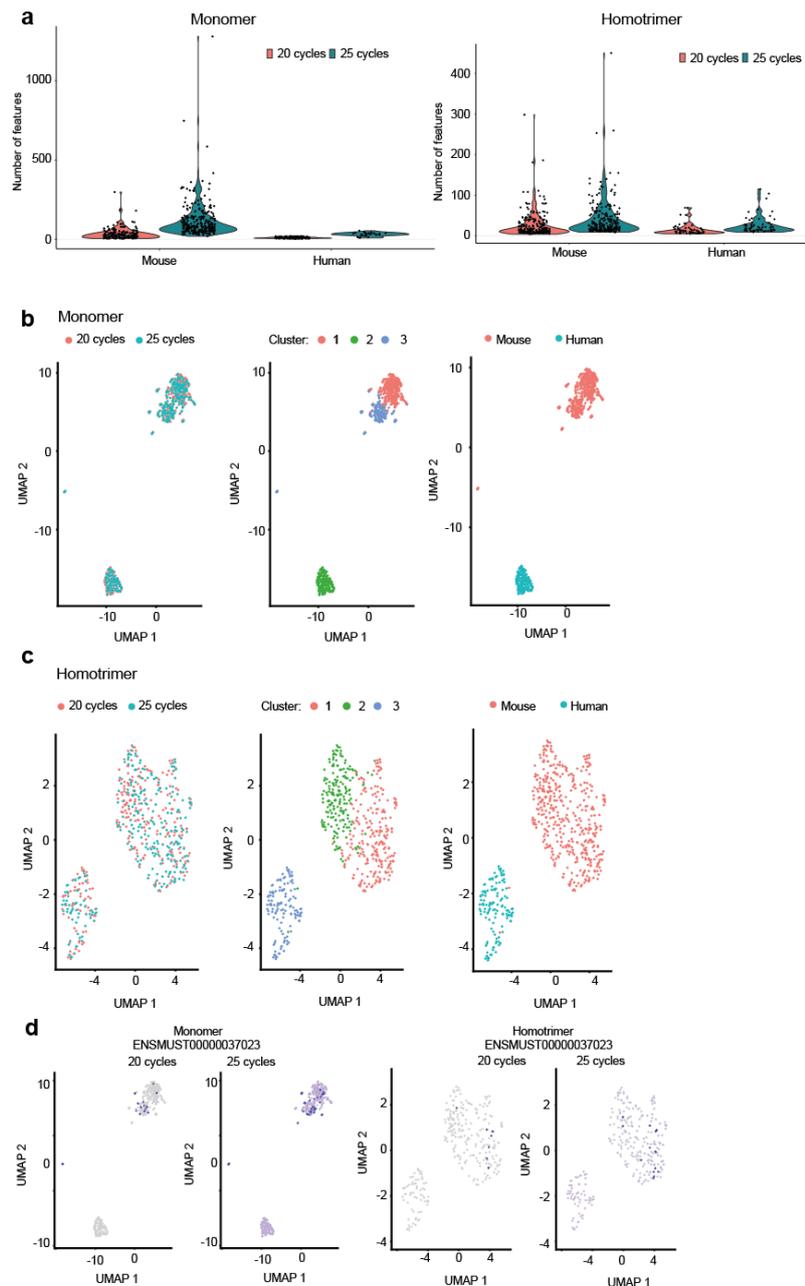
Supplementary Figure 17: Quality metrics for 10X chromium single-cell sequencing libraries amplified using 20 and 25 cycles of PCR.

The number of counts, features and number of UMIs for 10X Chromium libraries PCR amplified for 20 cycles (**a**) and 25 cycles (**b**). Each dot represents a single-cell following filtering. **c**, The log₁₀ genes per UMI plotted as a density.



Supplementary Figure 18: The number of discordant bases across the full length of the homotrimer UMI

The number of errors per read between the sequenced CMI and the ground truth CMI following 30 and 35 PCR cycles. Error bars are s.d. of 3 independent experiments.



Supplementary Figure 19: Number of features and UMAP clustering for the integrated analysis of homotrimer drop-seq UMIs following 20 and 25 cycles of PCR.

a, The number of features detected within the mouse and human cells for monomer (left panel) and homotrimer (right panel) UMIs following 20 and 25 PCR cycles. UMAP plots showing the integration, clustering and annotation of libraries amplified following 20 and 25 PCR cycles for monomer (**b**) and homotrimer (**c**) UMIs. **d**, UMAP plots showing the expression of a non-significant gene ENSMUST0000037023 in monomers (left panels) and homotrimer corrected (right panels) following 20 and 25 cycles of PCR. Even though this gene is not considered significantly different between 20 and 25 cycles in both the monomer and homotrimer datasets, there is generally an overall increase in background counts following 25 cycles of PCR in the monomer dataset that is not apparent within the homotrimer dataset.