

A new compression strategy to reduce the size of nanopore sequencing data

Kavindu Jayasooriya,^{1,2,3,4,7} Sasha P. Jenner,^{1,7} Pasindu Marasinghe,⁴ Udith Senanayake,⁴ Hassaan Saadat,⁵ David Taubman,⁵ Roshan Ragel,⁴ Hasindu Gamaarachchi,^{1,2,3,8} and Ira W. Deveson^{1,2,6,8}

¹Genomics and Inherited Disease Program, Garvan Institute of Medical Research, Sydney, New South Wales 2010, Australia;

²Centre for Population Genomics, Garvan Institute of Medical Research and Murdoch Children's Research Institute, Sydney, New South Wales 2010, Australia; ³School of Computer Science and Engineering, University of New South Wales, Sydney, New South Wales 2052, Australia; ⁴Department of Computer Engineering, University of Peradeniya, Peradeniya 20400, Sri Lanka; ⁵School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney, New South Wales 2052, Australia;

⁶St Vincent's Clinical School, Faculty of Medicine, University of New South Wales, Sydney, New South Wales 2052, Australia

Nanopore sequencing is an increasingly central tool for genomics. Despite rapid advances in the field, large data volumes and computational bottlenecks continue to pose major challenges. Here, we introduce ex-zd, a new data compression strategy that helps address the large size of raw signal data generated during nanopore experiments. Ex-zd encompasses both a lossless compression method, which modestly outperforms all current methods for nanopore signal data compression, and a 'lossy' method, which can be used to achieve additional savings. The latter component works by reducing the number of bits used to encode signal data. We show that the three least significant bits in signal data generated on instruments from Oxford Nanopore Technologies (ONT) predominantly encode noise. Their removal reduces file sizes by half without impacting downstream analyses, including basecalling and detection of modified DNA or RNA bases. Ex-zd compression saves hundreds of gigabytes on a single ONT sequencing experiment, thereby increasing the scalability, portability, and accessibility of nanopore sequencing.

[Supplemental material is available for this article.]

Nanopore sequencing enables high-throughput sequencing of native DNA or RNA molecules of any length. Platform updates from Oxford Nanopore Technologies (ONT) have enabled increasingly cost-effective and scalable sequencing in recent years (Wang et al. 2021; Marx 2023). As the technology continues to improve, there is a need for ongoing improvement in data management, storage, and analysis methods to match.

An ONT device measures the displacement of ionic current as a DNA or RNA molecule passes through a nanoscale protein pore. Time-series current signal data is recorded and 'basecalled' into sequence reads and can be analyzed directly to identify 'modified' DNA (Simpson et al. 2017; Zhang et al. 2023) or RNA (Jain et al. 2022) bases, DNA damage (An et al. 2015), RNA secondary structures (Bizuyehu et al. 2022; Stephenson et al. 2022), or other features beyond the primary nucleotide sequence (Wan et al. 2022). Because algorithms for ONT basecalling and other signal-level analysis processes are continually evolving, it is common practice to retain raw signal data for future re-analysis (Wan et al. 2022). Raw data retention is also critical for reproducibility, standardization, and open science.

We previously introduced a new file format for the storage and analysis of nanopore raw signal data called SLOW5 (and its bi-

nary equivalent BLOW5), one benefit of which was an average ~25% smaller file size compared to ONT's original native file format called FAST5 (Gamaarachchi et al. 2022). This reduction was achieved by addressing metadata redundancy and inefficient space allocation, and similar improvements were subsequently adopted by ONT in a new file format for signal data called POD5 (<https://github.com/nanoporetech/pod5-file-format>). BLOW5 and POD5 also employ similar lossless data compression methods, which reduce the size of the chain of sequential signal values that make up a raw nanopore read. Despite these savings, signal data in both formats remain ~10× larger than their corresponding basecalled reads, or ~1.7 TiB for a typical human genome sample at ~40× coverage (Supplemental Table S1).

The large size of raw ONT signal data creates several challenges. Long-term storage is expensive; a major consideration both for ONT users and for government-funded data repositories. Upload, download, or transfer of signal data sets is slow, may incur large egress costs, and is often not feasible in low-bandwidth settings, such as field studies or remote clinical sites. Large file sizes also create analysis bottlenecks, as data typically needs to be colocated with computer resources during the execution of analysis software, or even during sequencing, as data production on an ONT sequencing device rapidly consumes all disk space on the accompanying computer.

To alleviate these challenges, we have developed a new nanopore signal data compression strategy called ex-zd, which delivers

⁷Joint first authors.

⁸Joint senior authors.

Corresponding authors: hasindu@garvan.org.au, i.deveson@garvan.org.au

Article published online before print. Article, supplemental material, and publication date are at <https://www.genome.org/cgi/doi/10.1101/gr.280090.124>. Freely available online through the *Genome Research* Open Access option.

© 2025 Jayasooriya et al. This article, published in *Genome Research*, is available under a Creative Commons License (Attribution 4.0 International), as described at <http://creativecommons.org/licenses/by/4.0/>.

further space savings over existing methods. In doing so, we demonstrate how ONT signal data is amenable to ‘lossy’ data compression methods (Pu 2006), in which a portion of data is removed to greatly reduce file size with no impact on the utility of the data. We provide new ex-zd lossless and lossy compression methods for the nanopore community, via our open source libraries *slow5lib*, *pyslow5*, and data toolkit *slow5tools* (Samarakoon et al. 2023b).

Results

Lossless data compression with ex-zd

We developed a new compression strategy, called ex-zd, with the goal of improving nanopore signal data file sizes. Ex-zd can be used, among several alternate compression methods supported in *slow5lib*, *pyslow5*, and *slow5tools* (version 1.3.0 or later) (Samarakoon et al. 2023b), to reduce the size of data stored in

BLOW5 format (Gamaarachchi et al. 2022). Ex-zd compresses the chain of sequential signal data values that make up a read and should therefore be equally applicable to raw data written in ONT’s FAST5 or POD5 format.

By default, ex-zd is a ‘lossless’ compression method, meaning data is identical following compression and subsequent decompression. The lossless component of ex-zd builds upon an existing method, called VBZ (https://github.com/nanoporetech/vbz_compression), which is the current state-of-the-art for ONT data compression. A key element of VBZ is the transformation of each chain of raw signal values into a chain of differences between sequential values. Because most adjacent values are of similar magnitude, the differences or ‘zigzag deltas’ are small compared to the raw values. Ex-zd extends this concept, taking advantage of the high density of one-byte zig-zag deltas, which are encoded verbatim and separately from the two-byte data to achieve further savings (see Methods).

To evaluate this strategy, we applied lossless ex-zd compression to a typical human genome ONT sequencing data set generated with current standard protocols (HG002-Prom5K Chr 22 subset) (see Supplemental Table S1). We compared the compression ratio achieved by ex-zd on this data set to a wide range of other possible lossless compression methods ($n = 44$), including VBZ. Ex-zd achieved the highest compression ratio (2.35) of any method tested (Fig. 1; Supplemental Table S2). This translated to a 2.23% reduction in file size for a BLOW5 file compressed with ex-zd when compared to VBZ, 2.35% when compared to a native POD5 file, or a savings of 39 GiB on a typical human genome sequencing data set (Supplemental Table S1). We also observed that ex-zd compression adds minimal additional overhead in terms of computational time and RAM usage (Supplemental Table S3).

Although ex-zd showed best-in-class performance, it produced a relatively modest savings over existing alternatives. Furthermore, based on the small differences observed between the best performing methods tested above (Fig. 1), we believe we are approaching the limit of what is practically achievable with lossless compression methods.

Lossy data compression with ex-zd

To further reduce the size of signal data, ex-zd combines a lossy compression method, which can be optionally applied prior to the lossless encoding described above. Lossy compression methods, in which some portion of the starting data is nonreversibly removed to reduce the footprint, are common in other domains, such as image or audio processing (Pu 2006). One previous study considered the potential utility of lossy compression for nanopore sequencing data, with promising results (Chandak et al. 2021). However, there is currently no usable implementation of a lossy compression method available to ONT users, and further exploration is warranted.

Ex-zd lossy compression uses a simple bit-reduction strategy, which was motivated by the following observations regarding ONT signal data properties. Signal data generated on an ONT PromethION instrument are currently recorded using 11 bits. When plotting a frequency distribution of current signal values in their native 11-bit format, the distribution is not smooth but characterized by sporadic spikes where the frequencies of adjacent values differ substantially (Fig. 2A). This pattern is observed on diverse data sets, such as data from distantly related species, and is unrelated to the k -mer frequency profile of the underlying

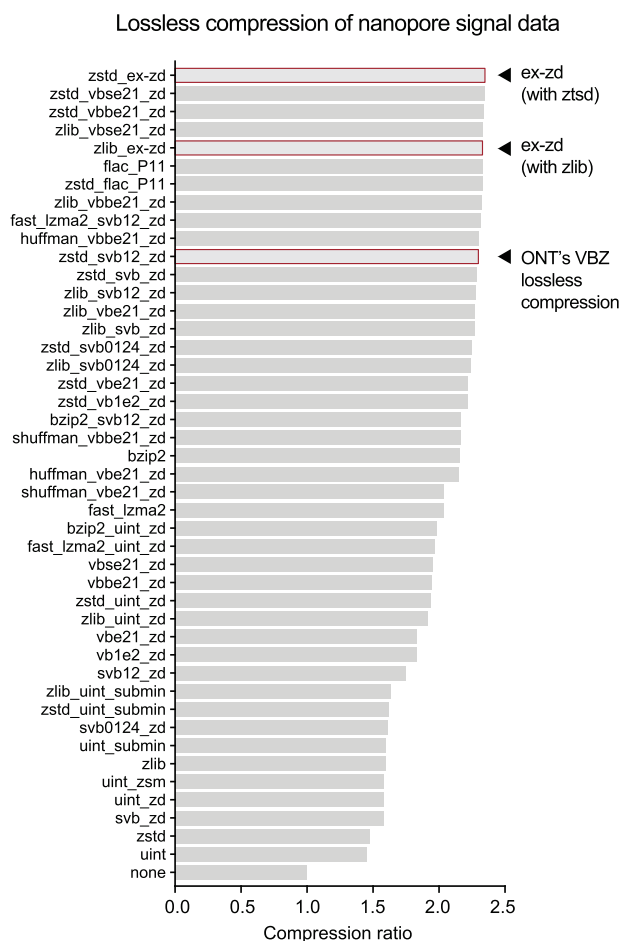


Figure 1. Comparison of alternative lossless compression methods. Bar chart shows compression ratios achieved when applying different lossless compression methods to a typical ONT PromethION signal data set (HG002-Prom5K Chr 22 subset) (see Supplemental Table S1). Compression ratio is calculated as follows: *Uncompressed Size/Compressed Size*. A wide range of alternative methods ($n = 44$) was tested, most of which combine multiple algorithms. Algorithms are indicated in shorthand with “_” separators on the vertical axis, and Supplemental Table S2 provides a full summary of the algorithms used.

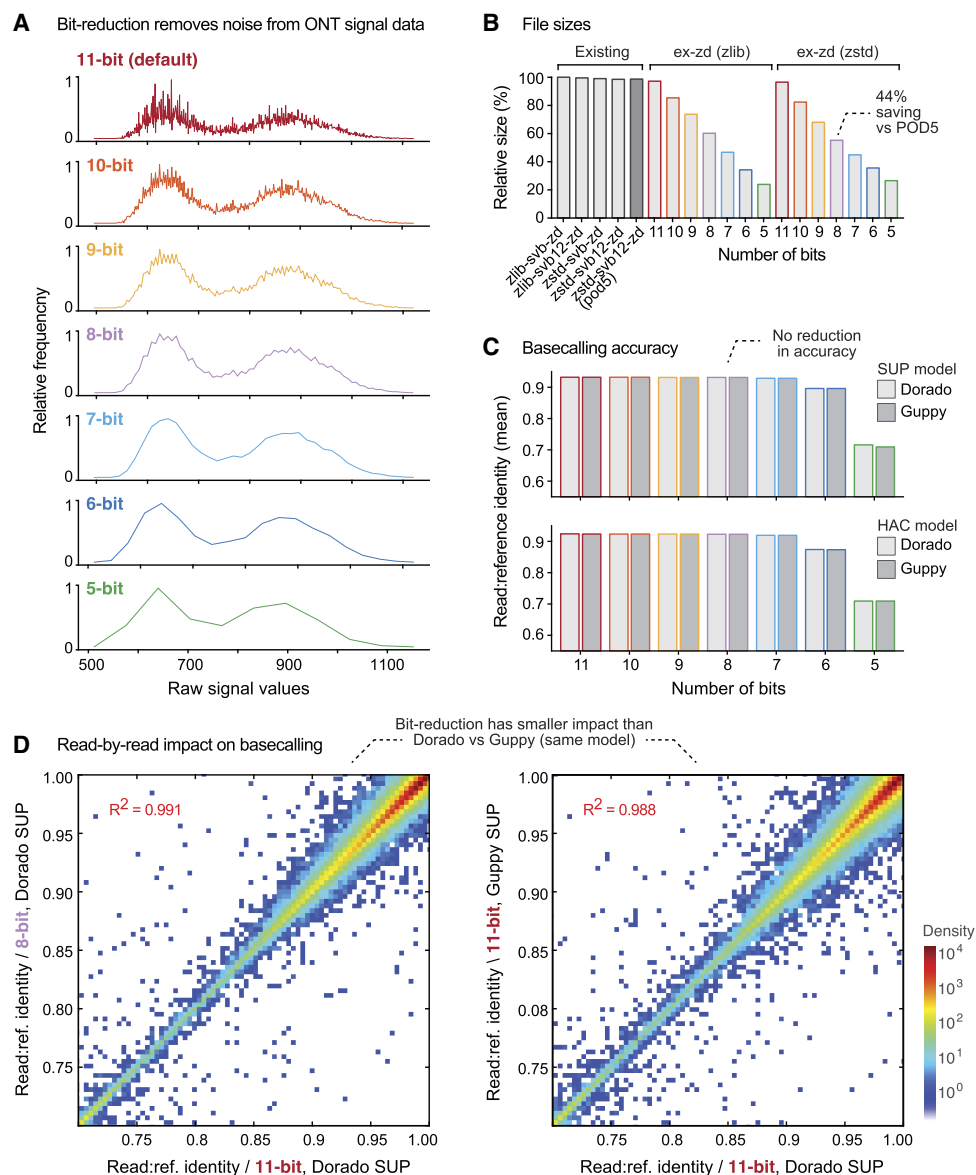


Figure 2. Evaluating ex-zd bit-reduction strategy for lossy compression of ONT PromethION data. (A) Frequency distributions for raw signal values in a typical ONT PromethION data set (HG002-Prom5K Chr 22 subset; see [Supplemental Table S1](#)) represented in native 11-bit encoding (red) or encoded with a smaller number of bits (10–5 bits). (B) Bar chart shows relative file sizes for the same data set in BLOW5 format with current lossless compression methods (gray bars) compared to lossy ex-zd compression with decreasing numbers of bits (11-bit down to 5-bit). Sizes are shown as percentages relative to zlib-sv-bz, which is currently the default compression method used in slow5tools/slow5lib. Native POD5 format, which uses zstd-svb12-zd (VBZ) compression, is shown for comparison. (C) Bar chart shows basecalling accuracy, as measured by mean read:reference identity, for the same data set and bit-reduced encodings as above. Basecalling accuracies are shown separately for ONT's Dorado (light gray) versus Guppy (dark gray) software and SUP (upper) versus HAC (lower) models. (D) Density scatterplots show read:reference identities for individual basecalled reads. The *left* plot compares native 11-bit data versus bit-reduced 8-bit data, both basecalled with Dorado SUP model. The *right* plot shows native 11-bit data basecalled with Guppy versus Dorado software, using the same SUP basecalling model.

sample ([Supplemental Fig. S1A](#)). Spikes occur reproducibly at specific signal values across independent reads and data sets and tend to occur on signal values when the two least significant bits of the values transition from 11_2 to 00_2 (e.g., 011_2 -to- 100_2 , 0111_2 -to- 1000_2 , etc.). It is highly unlikely that this unusual pattern reflects natural biomolecular and/or electrophysical dynamics at play during the sequencing process. It is more likely that this is an artifact of the analog to digital converter (ADC) or another hardware component used in ONT devices and could be erased without

compromising the molecular information encoded in the data. Importantly, we saw that this pattern of spikes was reduced when the same data set was represented with fewer than 11 bits, with a smooth bimodal frequency distribution obtained when data was encoded in just 7 bits (Fig. 2A).

This analysis suggests that the three or even four least significant bits in 11-bit signal data from an ONT PromethION primarily encode technical noise rather than useful signal. Therefore, file sizes may be reduced by decreasing the number of bits used to encode

signal values, without compromising the data. As an analogy, this is akin to reducing the number of decimal places used for each number when writing a list of numbers; fewer digits are required to produce the list, but there is little impact on the values encoded or the differences between successive values.

Prompted by these observations, we implemented a flexible bit-reduction strategy within ex-zd, in which the user can optionally reduce the number of bits used to encode signal values in a BLOW5 file from the default 11 bits for PromethION data down to 5 bits (or from 13 bits down to 7 bits for MinION data; see below). The N least significant bits are zeroed by rounding them down to 0 or up to 2^N , depending on which is closer. Ex-zd lossless compression is then applied to the bit-reduced data. The two methods are synergistic because bit-reduction increases the density of one-byte zigzag deltas, allowing the lossless algorithm to achieve higher compression ratios (see Methods). This results in significant reductions to the BLOW5 file size, with a >10% savings for each additional bit removed (Fig. 2B). For example, a BLOW5 file with 8-bit ex-zd compression is 44% smaller than native 11-bit POD5, or 737 GiB smaller for a human genome sequencing data set (Supplemental Table S1).

Validation of ex-zd lossy compression

It is critical that the space savings from ex-zd lossy compression do not come at the cost of data integrity. That is, we should see no meaningful impact on the outcomes of basecalling or other signal-level analysis when using bit-reduced data.

We first assessed the outcomes of ONT basecalling on a human genome sequencing data set encoded with decreasing numbers of bits, testing ONT's Dorado and Guppy basecalling software with both high accuracy (HAC) and super accuracy (SUP) models (see Methods). All analysis was performed on BLOW5 files, after confirming that the choice of file-type (BLOW5 vs. POD5 vs. FAST5) has no impact on basecalling outcomes (Supplemental Fig. S1B). By comparison to the 11-bit (i.e., lossless) encoding, we saw no reduction in basecalling accuracy for 10-bit, 9-bit, or 8-bit encoding, as assessed by mean, median, or modal read:reference identities (Fig. 2C). A small 0.3% mean reduction occurred at 7-bit, followed by a steep decline in basecalling accuracy when fewer than 7 bits were used (Fig. 2C; Supplemental Table S4). Similar trends were obtained during downstream variant detection using Clair3 (Zheng et al. 2022), with no reduction in variant accuracy for 10-bit, 9-bit, or 8-bit encodings (Supplemental Fig. S1C). Scatterplots showing read:reference identities for individual reads between data sets with different encodings indicated highly similar outcomes at 8-bit or above (Fig. 2D). Although not all reads are identical, major outliers generally resulted from divergent mapping positions for nonuniquely aligned reads rather than large discrepancies in their basecalled sequences (Supplemental Table S5). We observed a greater degree of difference between identical 11-bit data basecalled with Dorado versus Guppy software using the same underlying models ($R^2 = 0.988$) than between an 8-bit versus 11-bit ex-zd encoding ($R^2 = 0.991$) (Fig. 2D). The concordance between 8-bit versus 11-bit data was also equivalent to the comparison of identical 11-bit data basecalled with the same Dorado version using different GPU hardware (NVIDIA A100 vs. V100), and we confirmed the above results with both Dorado v4 and the most recent v5 models (Supplemental Fig. S2). Therefore, the small degree of difference seen in this read-level analysis reflects inherent stochasticity in the basecalling and alignment process, not a result of ex-zd

lossy compression, and is implicitly tolerated by the nanopore community.

We next considered the impact of ex-zd lossy compression on 5-methylcytosine (5mC) DNA methylation profiling. We assessed performance by comparison of 5mC frequencies at CpG sites ascertained by Dorado or Guppy on ONT data to matched reference data generated with whole-genome bisulphite sequencing (WGBS) (see Methods). We observed no reduction in the correlation of ONT versus WGBS results across global CpG sites for encodings of 8-bit or greater (Fig. 3A,B; Supplemental Table S6). As was observed for basecalling accuracy, individual reads showed highly similar methylation states between different encodings and a greater degree of difference between Dorado versus Guppy ($R^2 = 0.929$) than the 8-bit versus 11-bit encoding ($R^2 = 0.971$) (Fig. 3C). These results were recapitulated when using open source methylation profiling software f5c (Gamaarachchi et al. 2020) as an alternative to Guppy or Dorado (Supplemental Table S6). All basecalling and methylation profiling results were also recapitulated as above using a data set generated with a 4 KHz (rather than 5 KHz) data sampling rate, as was used on ONT devices prior to 2023, and with data generated with the previous generation of ONT flow cells (R9.4.1) (Supplemental Tables S7, S8).

Data generated on an ONT MinION device is natively encoded with 13 bits, rather than 11 bits for PromethION data. Using a typical MinION data set (HG002-Min5K) (see Supplemental Table S1), we next confirmed that ex-zd lossy compression is also effective on MinION data. We found that up to 3 bits could be removed with no impact on basecalling or 5mC profiling, delivering a space saving of 44% at a 10-bit versus 13-bit encoding (Supplemental Fig. S3A–D; Supplemental Table S9).

Finally, we assessed the suitability of ex-zd lossy compression on RNA sequencing data generated using ONT's RNA004 protocol (see Supplemental Table S1). Bit-reduction and ex-zd compression achieved equivalent space savings on RNA data to what was observed previously with DNA data above (Fig. 4A). Similarly, 3 bits could be removed with no impact on RNA basecalling accuracy down to a 3-bit reduction, as measured on native human mRNA (UHRR-Prom; sequenced on a PromethION) and synthetic RNA controls (SIRV-Min; sequenced on a MinION) (Fig. 4B). We assessed basecalling outcomes and detection of N⁶-Methyladenosine (m⁶A) modifications with m⁶Anet (Hendra et al. 2022), at the individual read level, and found 3-bit reduction had no greater impact than a minor version change in the Dorado basecalling software (Fig. 4C,D; Supplemental Table S10). In summary, we observed no meaningful impact in the quality of RNA basecalling or detection of RNA modifications (m⁶A) when applying ex-zd lossy compression with up to 37 bits removed.

Discussion

With the breadth of ONT sequencing adoption and the scale of data sets growing (Alonge et al. 2020; Beyter et al. 2021; Reis et al. 2023; Chen et al. 2025), there is a need for new and efficient methods for data storage and data sharing. Ex-zd is a new compression strategy that can be used to reduce file sizes of raw nanopore signal data to help address this challenge. Ex-zd encompasses both a lossless compression method, which modestly outperforms other available methods, and a lossy bit-reduction method, with the two working in tandem to deliver substantial savings.

Although lossy compression methods are popular in other domains and are already used in some areas of genomics (Illumina 2014; Bonfield 2022), they are not widely used for

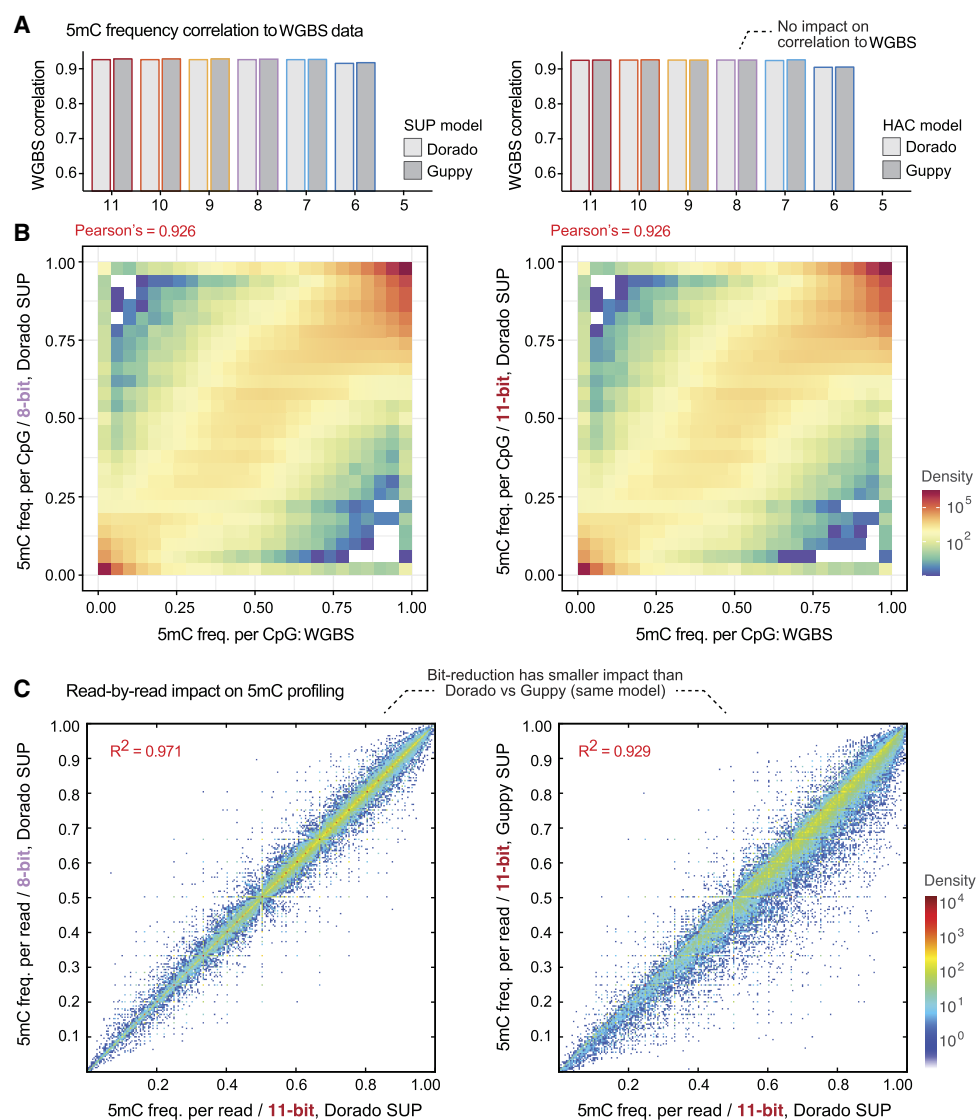


Figure 3. Impact of ex-zd bit-reduction on ONT DNA methylation profiling. (A) Bar chart shows the correlation of 5mC methylation frequencies at global CpG sites recorded with whole-genome bisulfite sequencing (WGBS) versus ONT methylation profiling on a matched sample (HG002). ONT methylation profiling was performed with either Dorado (light gray) versus Guppy (dark gray) software and SUP (left) versus HAC (right) models on the same raw signal data set (HG002-Prom5K Chr 22 subset) (see Supplemental Table S1) encoded with a decreasing number of bits (native 11-bit down to 5-bit). (B) Density scatterplots compare global 5mC profiles from native 11-bit data and bit-reduced 8-bit data to WGBS data, as per the above comparisons. (C) Density scatterplots show 5mC methylation frequencies for individual basecalled reads, that is, the fraction of CpG bases within a given read that are called as being methylated. The left plot compares native 11-bit data versus bit-reduced 8-bit data, both basecalled with Dorado SUP model. The right plot shows native 11-bit data basecalled with Guppy versus Dorado software, using the same SUP methcalling model.

nanopore signal data compression. Lossy methods irreversibly transform the underlying data and are generally avoided in scenarios where it is more important to maximize precision than to reduce the storage footprint of the data (Pu 2006). However, we demonstrate above that ONT PromethION signal data can be reduced from 11-bit to 8-bit encoding with no negative impact on analysis outcomes for either basecalling or detection of modified bases (e.g., 5mC), thereby delivering space savings without a trade-off in precision. In fact, our analyses indicate that the three least or even four significant bits in native ONT data primarily encode noise. Given that 8-bit PromethION data with ex-zd compression is ~45% smaller than 11-bit native POD5 format, this is an important development for the field. Moreover, this provides the basis

to evaluate and/or develop alternative lossless or lossy compression strategies, which may be applied on top of bit-reduction to deliver greater savings. For example, our preliminary observations suggest the Free Lossless Audio Codec (FLAC) algorithm, commonly used for audio-compression, may be well suited for compression of bit-reduced ONT signal data (see Supplemental Note 1). Although our results demonstrate the promise of lossy compression methods for nanopore data, any lossy method must be rigorously evaluated and applied with care, as their misuse can permanently compromise the user's data.

Our results show equivalent basecalling accuracy with bit-reduced 8-bit PromethION data compared to native 11-bit and just a small (0.3%) reduction in accuracy with 7-bit data. It is interesting

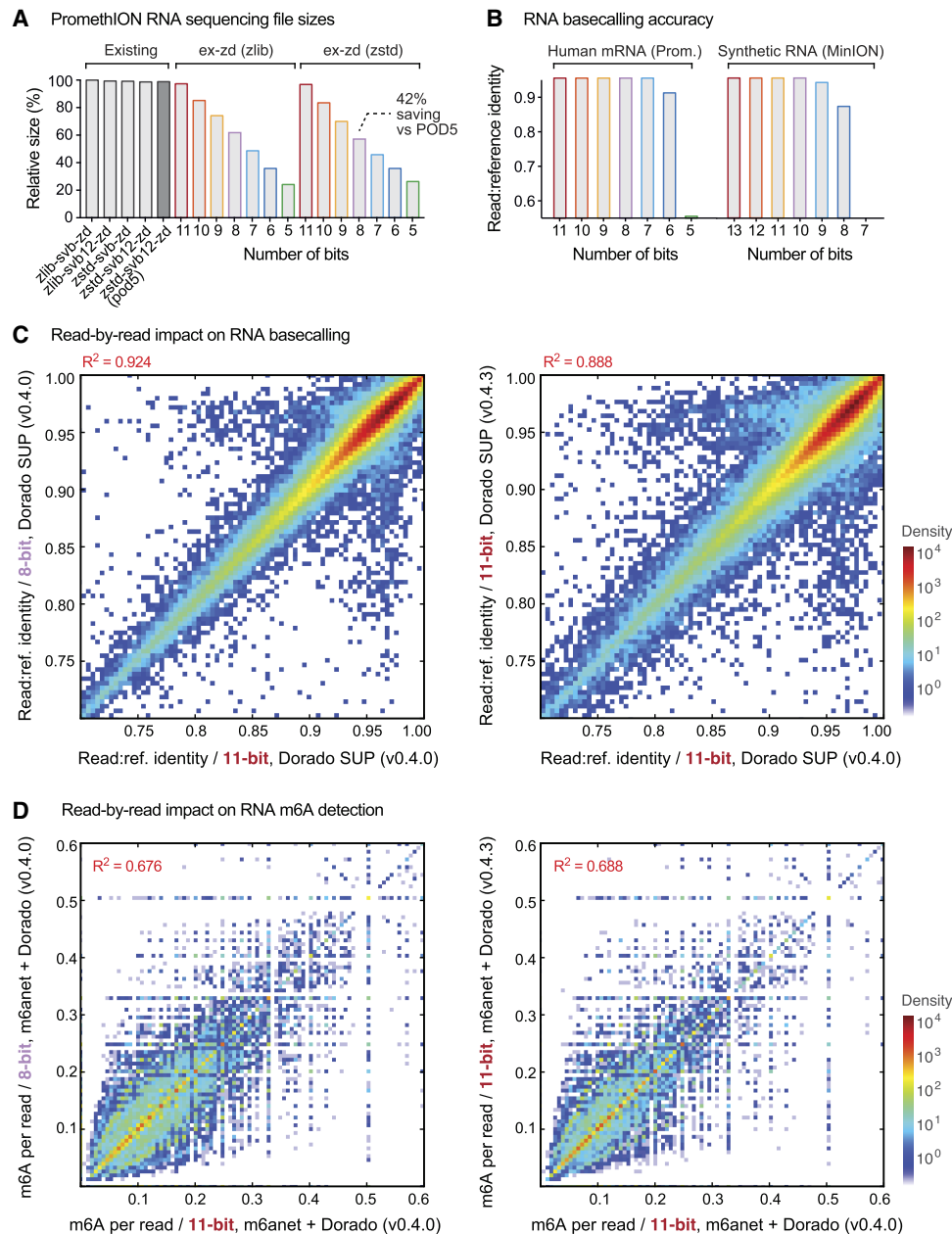


Figure 4. Suitability of ex-zd bit-reduction for ONT direct RNA sequencing data. (A) Bar chart shows relative file sizes for a typical ONT PromethION data set generated with the RNA004 sequencing kit with current lossless compression methods (gray bars) compared to lossy ex-zd compression with data encoded with a decreasing number of bits (11-bit down to 5-bit). Sizes are shown as percentages relative to svb-zd-zlib, which is currently the default compression method used in slow5tools/slow5lib. Native POD5 format, which uses svb12-zd-zstd (VBZ) compression, is shown for comparison. (B) Bar chart shows Dorado SUP basecalling accuracy, as measured by mean read:reference identity, for the same bit-reduced encodings as in A. Basecalling accuracies are shown separately for a human mRNA sample sequenced on PromethION (UHRR-Prom) and for SIRV synthetic RNA controls sequenced on a MinION (SIRV-Min) (see Supplemental Table S1). (C) Density scatterplots show read:reference identities for individual basecalled reads from the UHRR-Prom data set above. The left plot compares native 11-bit data versus bit-reduced 8-bit data, both basecalled with Dorado SUP model. The right plot shows native 11-bit data basecalled with Dorado SUP model, using two different but closely matched Dorado software versions (0.4.0 vs. 0.4.3). (D) For the same data sets as in C, scatterplots show m6A methylation frequencies for individual reads, that is, the fraction of 'A' bases within a given read that are called as being m6A by the m6Anet software.

to note that the ONT basecalling models used here are neural network models, trained on 11-bit data. Given the characteristic differences seen between 11-bit versus 7-bit data (see Fig. 2A), we were surprised at the strong performance on 8-bit and 7-bit data. This opens the intriguing possibility that basecalling performance could be improved via re-training on bit-reduced data. We hypoth-

esize that the removal of noise from the signal data, which appears to be optimal for the 7-bit encoding, may have analytic advantages.

File size reductions delivered by ex-zd or other future lossless methods will have many benefits for the community. The most obvious will be proportional reductions in the cost of data storage,

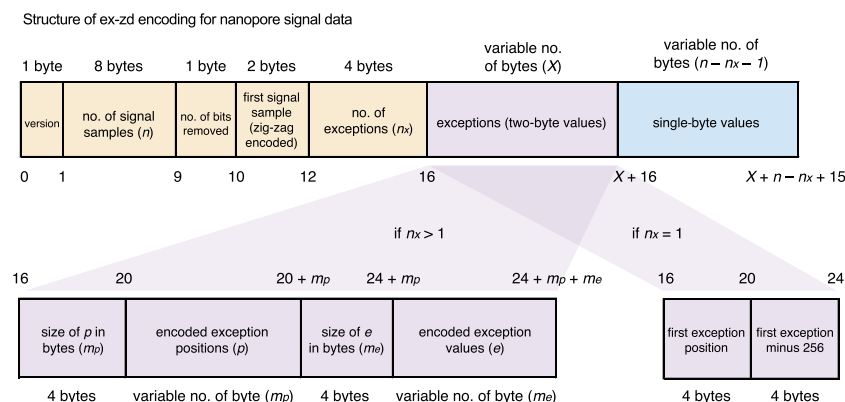


Figure 5. Schematic overview of ex-zd lossless compression strategy. Schematic illustrates the structure of the raw signal values for a single nanopore sequencing read encoded with ex-zd. Orange blocks represent the ex-zd metadata; blue block represents one-byte data; purple blocks represent two-byte exception data. Assuming exceptions exist, the exception data structure can take two forms (shown below), depending on whether the number of exceptions is $n_x > 1$ or $n_x = 1$. If there are no exceptions, $X = 0$ (i.e., purple block is absent).

which are a major expense both for everyday users and for public data repositories, such as EBI's European Nucleotide Archive (ENA) or NCBI's Sequence Read Archive (SRA). The time and cost required to upload/download data from these repositories will be similarly reduced, encouraging open data sharing of raw signal data. This complements our recent tool *slowScurl* (Wong et al. 2024) which allows a user to quickly fetch specific reads (e.g., for a gene of interest) from a nanopore signal data set on a remote server, such as ENA or SRA, without downloading the entire data set. Smaller file sizes will facilitate data transfer between sites with limited bandwidth, which can be a major obstacle for remote field studies enabled by portable ONT devices (Quick et al. 2016). The less obvious impact of file size reductions will be to increase sequencing throughput on ONT devices, such as the PromethION P48, where available storage can currently accommodate only around half of the maximum theoretical data generation capacity. Applying ex-zd compression to each new batch of reads generated during sequencing would increase the sequencing throughput that is practically achievable by almost two-fold (given the 44% space saving with 3-bit reduction), without any further updates to the hardware. Finally, smaller file sizes can also address a common analysis bottleneck for ONT users, wherein disk space required to hold data during analysis is the limiting resource, rather than computer capacity. In such a scenario, a pedantic user may choose to apply lossy compression to their data set to alleviate space constraints during analysis, while retaining an original lossless copy in their archive for long-term storage.

Ex-zd is the latest innovation in the SLOW5 data ecosystem (<https://hasindu2008.github.io/slow5/>), which includes the SLOW5/BLOW5 file format itself (Gamaarachchi et al. 2022); software libraries for reading/writing files (<https://github.com/hasindu2008/slow5lib>); a toolkit for working with SLOW5/BLOW5 files (Samarakoon et al. 2023b); the *slowScurl* utility for remote data access (Wong et al. 2024); BLOW5-enabled basecalling software (Samarakoon et al. 2023a); packages for simulation (Gamaarachchi et al. 2024) and visualization (Samarakoon et al. 2024) of signal data; and a range of other open source tools (Simpson et al. 2017; Gamaarachchi et al. 2020; Zhang et al. 2021; Shih et al. 2022; Senanayake et al. 2023; Firtina et al. 2024; Guo et al. 2024; Kovaka et al. 2025). Ex-zd compression is now sup-

ported within *slowlib*, *pyslow5*, and *slow5tools*, and all methods and formats are open source, in case ONT or other future nanopore vendors want to adopt them.

Methods

Ex-zd compression strategy

Ex-zd is a new compression strategy for nanopore signal data, which separately encodes one-byte and two-byte zig-zag delta transformed data. The ex-zd strategy is illustrated in Figure 5, and mathematical derivations are provided in Supplemental Note 2.

The ex-zd encoding begins by writing the version number using one byte, followed by the number of signal samples written using eight bytes, then the number of bits eliminated during the lossy encoding using one byte (Fig. 5).

Next, each signal sample is bit-shifted to the right by the smallest length of successive zero least significant bits (which is greater than or equal to the number of bits eliminated during lossy compression). Next, the zig-zag delta transformation is applied. In this transformation, the first signal sample followed by the consecutive differences (deltas) are zig-zag encoded, meaning positive integers are doubled and the absolute value of negative integers are doubled then subtracted by one. The first signal sample after zig-zag encoding is then written using two bytes (Fig. 5). Afterwards, the data is divided into two groups: integers which fit into one byte (the one-byte values) and those which require two bytes (the exceptions). The exceptions are now subtracted by 256 (256 is the minimum value that an exception can have). The number of exceptions is written using four bytes (Fig. 5). If there is only one exception, the exception's position and the exception are both written using four bytes each. When there are zero exceptions, no exception data is written (the purple box in Fig. 5 would not exist). If there is more than one exception, the positions of the exceptions are encoded as follows: the first position is left unchanged, while the remainder are delta encoded and subtracted by 1; finally, all the integers are streamvbyte-encoded. The size of this encoding is written using four bytes, followed by the encoding itself. Next, the exceptions are streamvbyte-encoded. As before, the size of this encoding is written using four bytes, followed by the encoding itself. Finally, each data point in the one-byte data is written using one byte (blue box in Fig. 5).

Bit elimination during ex-zd lossy compression

Ex-zd lossy compression is based on a simple bit-reduction strategy, in which the user can specify the number of bits to be eliminated from their signal data set. If n bits are to be eliminated, for each signal value x , the following bit-wise rounding operation is applied that will zero the n least significant bits:

$$x \& \sim (2^n - 1) + 2^n(x \& 2^{n-1}).$$

When performing the bit-reduction, the number of bits eliminated is stored as described above and the signal values are bit-shifted to the right. When decoding, the values are left-shifted by this same amount. During lossless encoding, this will be zero and no shifting is performed.

Benchmark experiments

Data sets

The data sets used for the experiments are listed in [Supplemental Table S1](#). HG002-Prom5K is a DNA sequencing experiment run on the popular human genome reference sample HG002, sequenced on an ONT PromethION device with a R10.4.1 flowcell, and the data was collected at 5 kHz sampling rate. HG002-Prom4K is similar except that the data was collected at 4 kHz. HG002-Min5K is sequenced on a MinION R10.4.1 at 5 kHz. UHRR-Prom is a direct RNA sequencing experiment run on the human transcriptome reference sample, Universal Human Reference RNA (Agilent). This was sequenced on a PromethION using the latest RNA004 kit and flowcell for direct RNA sequencing. Similar HG001 and UUHR data sets were also available from the previous generation R9.4.1 PromethION flowcell version. For many experiments, a limited subset of the full data set was used to minimize computer resources. This was achieved by subsetting reads corresponding to human Chr 22. Subsets were generated by basecalling the signal data, aligning the reads to the hg38 reference using minimap2 (Li 2018) and then extracting those reads using slow5tools (Samarakoon et al. 2023b). The 500 K subset was generated by randomly picking 500,000 reads from the signal data set.

All data sets above are publicly available. The large data sets HG002-Prom5K, HG002-Prom4K, and UHRR-Prom used for benchmarking experiments are available at the European Nucleotide Archive under accession number PRJEB64652 (Runs ERR12997168, ERR11777845, and ERR12997170, respectively). The smaller data sets HG002-Prom5K (Chr 22 subset), HG002-Prom4K (Chr 22 subset), HG002-Min5K, UHRR-Prom (500K read subset), HG001-PromR9 (Chr 22 subset), and UHRR-PromR9 are available through the Dryad data set (<https://doi.org/10.5061/dryad.1vvhm3p>). Direct RNA data on SIRV synthetic controls (SIRV-Min) are available through Zenodo (<https://zenodo.org/records/14676368>). Bisulphite data was downloaded from publicly available sources: for HG001 from ENCODE (ENCFF835NTC) and for HG002 from ONT open-data AWS repository (s3://ont-open-data/gm24385_mod_2021.09/bisulphite/cpg).

File sizes and performance measurements

The experiments for measuring the file sizes and performance were executed on a server with an Intel Xeon Silver 4114 CPU (20 cores, 40 threads), 376 GiB RAM, and an HDD-based network-attached storage (12 spinning disks configured with RAID 10) mounted via Network File System (NFS). The system was running Ubuntu 18.04.5 as the operating system. File sizes were measured using the `du` command ([Supplemental Note 3](#)). The runtime and peak RAM were measured using GNU time utility. Converting to/from lossless ex-zd was performed using slow5tools v1.3.0 view. Lossy compression was performed using slow5tools degrade. The disk I/O cache (pagecache, dentries, and inodes) was cleaned before runtime measurement experiments. Details of the commands and software versions are in [Supplemental Note 3](#).

Accuracy evaluation

Basecalling and 5mC calling were performed using Guppy (via butterfly-eel) (Samarakoon et al. 2023a) and Dorado (via slow5-dorado; <https://github.com/hiruna72/slow5-dorado>), with full commands and versions provided in [Supplemental Note 3](#). Basecalled reads were aligned to the reference (hg38 with no alternate contigs for DNA data and GENCODE v40 human transcriptome for RNA data) using minimap2. For measuring the basecalling accuracies, blast-like identity scores were calculated for primary alignments

using paftools.js in the minimap2 package (blast-like identity score = 10th column divided by 11th column in a PAF file). To measure the 5mC calling accuracy, we first mapped the basecalls with methylation tags using minimap2, sorted them using SAMtools (Li et al. 2009), and then the methylation frequencies were extracted using modkit v0.1.13 ([Supplemental Note 3](#)). The 5mC methylation frequencies were compared to publicly available data from whole-genome bisulfite sequencing (see Methods) using the compare_methylation.py script associated with nanopolish/f5c (Gamaarachchi et al. 2020). To assess per-read 5mC calling, we extracted the modification calls per site using modkit extract ([Supplemental Note 3](#)). Then, we extracted the modification type of interest (mod_code 'm' for 5mC). Then, per each read, we calculated the modification frequency across the read, taking modification probability >0.8 as 'modified' and <0.2 as 'unmodified.' The modification frequency of a given read was calculated as modified calls/(modified calls + unmodified calls). For obtaining m6A calls for RNA, we used m6anet followed by f5c eventalign ([Supplemental Note 3](#)). For calculating the per-read methylation frequency, we took modification probability >0.033 as 'modified' or otherwise 'unmodified.'

Software availability

Ex-zd compression implementation is available through slow5lib (<https://github.com/hasindu2008/slow5lib>) and slow5tools (<https://github.com/hasindu2008/slow5tools>) version 1.3.0 onwards. Slow5tools version 1.3.0 is also provided as [Supplemental Code](#). A detailed user guide on safe usage of ex-zd lossy compression is provided at GitHub (<https://hasindu2008.github.io/slow5tools/archive-lossy.html>).

Competing interest statement

I.W.D. manages a fee-for-service sequencing facility at the Garvan Institute of Medical Research and is a customer of Oxford Nanopore Technologies but has no further financial relationship. H.G. and I.W.D. have previously received travel and accommodation expenses from Oxford Nanopore Technologies. I.W.D. has paid consultant roles with Sequin PTY. H.G. has paid consultant roles with Sequin PTY and Swan Genomics PTY. The authors declare no other competing financial or nonfinancial interests.

Acknowledgments

We acknowledge the following funding support: Australian Medical Research Futures Fund grants 2016008 and 2023126 (to I.W.D.), National Health and Medical Research Council (NHMRC) grant 2035037 (to I.W.D.), and Australian Research Council (ARC) DECRA Fellowship DE230100178 and ARC Discovery Project DP230100651 (to H.G.). The views expressed herein are those of the authors and are not necessarily those of the Australian Government or the ARC, NHMRC, or MRFF. We thank Sri Parameswaran and John Stavrakakis (University of Sydney) for insightful discussions. We also thank James Ferguson (Garvan Institute) for assistance with butterfly-eel and pyslow5.

Author contributions: All authors contributed to the conception, design, and benchmarking of ex-zd. K.J., S.P.J., and H.G. implemented ex-zd and integrated into slow5lib and slow5tools. K.J., S.P.J., and H.G. performed benchmarking experiments. S.P.J., H.G., and I.W.D. prepared the figures and manuscript.

References

- Alonge M, Wang X, Benoit M, Soyk S, Pereira L, Zhang L, Suresh H, Ramakrishnan S, Maumus F, Ciren D, et al. 2020. Major impacts of widespread structural variation on gene expression and crop improvement in tomato. *Cell* **182**: 145–161.e23. doi:10.1016/j.cell.2020.05.021
- An N, Fleming AM, White HS, Burrows CJ. 2015. Nanopore detection of 8-oxoguanine in the human telomere repeat sequence. *ACS Nano* **9**: 4296–4307. doi:10.1021/acs.nano.5b00722
- Beyter D, Ingimundardottir H, Oddsson A, Eggertsson HP, Björnsson E, Jonsson H, Atlason BA, Kristmundsdottir S, Mehringer S, Hardarson MT, et al. 2021. Long-read sequencing of 3,622 Icelanders provides insight into the role of structural variants in human diseases and other traits. *Nat Genet* **53**: 779–786. doi:10.1038/s41588-021-00865-4
- Bizuayehu TT, Labun K, Jakubec M, Jefimov K, Niazi AM, Valen E. 2022. Long-read single-molecule RNA structure sequencing using nanopore. *Nucleic Acids Res* **50**: e120. doi:10.1093/nar/gkac775
- Bonfield JK. 2022. CRAM 3.1: advances in the CRAM file format. *Bioinformatics* **38**: 1497–1503. doi:10.1093/bioinformatics/btac010
- Chandak S, Tatwawadi K, Sridhar S, Weissman T. 2021. Impact of lossy compression of nanopore raw signal data on basecalling and consensus accuracy. *Bioinformatics* **36**: 5313–5321. doi:10.1093/bioinformatics/btaa1017
- Chen Y, Davidson NM, Wan YK, Yao F, Su Y, Gamaarachchi H, Sim A, Patel H, Low HM, Hendra C, et al. 2025. A systematic benchmark of Nanopore long-read RNA sequencing for transcript-level analysis in human cell lines. *Nat Methods* **22**: 801–812. doi:10.1038/s41592-025-02623-4
- Firtina C, Soysal M, Lindegger J, Mutlu O. 2024. RawHash2: mapping raw nanopore signals using hash-based seeding and adaptive quantization. *Bioinformatics* **40**: btac478. doi:10.1093/bioinformatics/btac478
- Gamaarachchi H, Lam CW, Jayatilaka G, Samarakoon H, Simpson JT, Smith MA, Parameswaran S. 2020. GPU accelerated adaptive banded event alignment for rapid comparative nanopore signal analysis. *BMC Bioinformatics* **21**: 343. doi:10.1186/s12859-020-03697-x
- Gamaarachchi H, Samarakoon H, Jenner SP, Ferguson JM, Amos TG, Hammond JM, Saadat H, Smith MA, Parameswaran S, Deveson IW. 2022. Fast nanopore sequencing data analysis with SLOW5. *Nat Biotechnol* **40**: 1026–1029. doi:10.1038/s41587-021-01147-4
- Gamaarachchi H, Ferguson JM, Samarakoon H, Liyanage K, Deveson IW. 2024. Simulation of nanopore sequencing signal data with tunable noise parameters. *Genome Res* **34**: 778–783. doi:10.1101/gr.278730.123
- Guo Z, Ni Y, Tan L, Shao Y, Ye L, Chen S, Li R. 2024. Nanopore Current Events Magnifier (nanoCEM): a novel tool for visualizing current events at modification sites of nanopore sequencing. *NAR Genom Bioinform* **6**: lqae052. doi:10.1093/nargab/lqae052
- Hendra C, Pratanwanich PN, Wan YK, Goh WSS, Thierry A, Göke J. 2022. Detection of m6A from direct RNA sequencing using a multiple instance learning framework. *Nat Methods* **19**: 1590–1598. doi:10.1038/s41592-022-01666-1
- Illumina. 2014. *Reducing whole-genome data storage footprint*. Illumina, Inc, San Diego. http://www.illumina.com/documents/products/whitepapers/whitepaper_datacompression.pdf.
- Jain M, Abu-Shumays R, Olsen HE, Akeson M. 2022. Advances in nanopore direct RNA sequencing. *Nat Methods* **19**: 1160–1164. doi:10.1038/s41592-022-01633-w
- Kovaka S, Hook PW, Jenike KM, Shivakumar V, Morina LB, Razaghi R, Timp W, Schatz MC. 2025. Uncalled4 improves nanopore DNA and RNA modification detection via fast and accurate signal alignment. *Nat Methods* **22**: 681–691. doi:10.1038/s41592-025-02631-4
- Li H. 2018. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**: 3094–3100. doi:10.1093/bioinformatics/bty191
- 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**: 2078–2079. doi:10.1093/bioinformatics/btp352
- Marx V. 2023. Method of the year: long-read sequencing. *Nat Methods* **20**: 6–11. doi:10.1038/s41592-022-01730-w
- Pu IM. 2006. Fundamental data compression. In *Digital Video and HD*, 2nd ed., pp. 1–17. Elsevier, Amsterdam. doi:10.1016/B978-075066310-6/50004-0 [accessed August 19, 2024].
- Quick J, Loman NJ, Duraffour S, Simpson JT, Severi E, Cowley L, Bore JA, Koundouno R, Dudas G, Mikhail A, et al. 2016. Real-time, portable genome sequencing for Ebola surveillance. *Nature* **530**: 228–232. doi:10.1038/nature16996
- Reis ALM, Rapadas M, Hammond JM, Gamaarachchi H, Stevanovski I, Kumaheri MA, Chintalaphani SR, Dissanayake DSB, Siggs OM, Hewitt AW, et al. 2023. The landscape of genomic structural variation in Indigenous Australians. *Nature* **624**: 602–610. doi:10.1038/s41586-023-06842-7
- Samarakoon H, Ferguson JM, Gamaarachchi H, Deveson IW. 2023a. Accelerated nanopore basecalling with SLOW5 data format. *Bioinformatics* **39**: btad352. doi:10.1093/bioinformatics/btad352
- Samarakoon H, Ferguson JM, Jenner SP, Amos TG, Parameswaran S, Gamaarachchi H, Deveson IW. 2023b. Flexible and efficient handling of nanopore sequencing signal data with *slow5tools*. *Genome Biol* **24**: 69. doi:10.1186/s13059-023-02910-3
- Samarakoon H, Liyanage K, Ferguson JM, Parameswaran S, Gamaarachchi H, Deveson IW. 2024. Interactive visualisation of raw nanopore signal data with *Squiggliser*. *Bioinformatics* **40**: btac501. doi:10.1093/bioinformatics/btac501
- Senanayake A, Gamaarachchi H, Herath D, Ragel R. 2023. DeepSelectNet: deep neural network based selective sequencing for Oxford nanopore sequencing. *BMC Bioinformatics* **24**: 31. doi:10.1186/s12859-023-05151-0
- Shih PJ, Saadat H, Parameswaran S, Gamaarachchi H. 2022. Efficient real-time selective genome sequencing on resource-constrained devices. *GigaScience* **12**: giad046. doi:10.1093/gigascience/giad046
- Simpson JT, Workman RE, Zuzarte PC, David M, Dursi LJ, Timp W. 2017. Detecting DNA cytosine methylation using nanopore sequencing. *Nat Methods* **14**: 407–410. doi:10.1038/nmeth.4184
- Stephenson W, Razaghi R, Busan S, Weeks KM, Timp W, Smibert P. 2022. Direct detection of RNA modifications and structure using single-molecule nanopore sequencing. *Cell Genom* **2**: 100097. doi:10.1016/j.xgen.2022.100097
- Wan YK, Hendra C, Pratanwanich PN, Göke J. 2022. Beyond sequencing: machine learning algorithms extract biology hidden in nanopore signal data. *Trends Genet* **38**: 246–257. doi:10.1016/j.tig.2021.09.001
- Wang Y, Zhao Y, Bollas A, Wang Y, Au KF. 2021. Nanopore sequencing technology, bioinformatics and applications. *Nat Biotechnol* **39**: 1348–1365. doi:10.1038/s41587-021-01108-x
- Wong B, Ferguson JM, Do JY, Gamaarachchi H, Deveson IW. 2024. Streamlining remote nanopore data access with *slow5curl*. *GigaScience* **13**: giae016. doi:10.1093/gigascience/giae016
- Zhang H, Li H, Jain C, Cheng H, Au KF, Li H, Aluru S. 2021. Real-time mapping of nanopore raw signals. *Bioinformatics* **37**: i477–i483. doi:10.1093/bioinformatics/btab264
- Zhang Y, Zhang Q, Yang X, Gu X, Chen J, Shi T. 2023. 6mA DNA methylation on genes in plants is associated with gene complexity, expression and duplication. *Plants* **12**: 1949. doi:10.3390/plants12101949
- Zheng Z, Li S, Su J, Leung AW-S, Lam T-W, Luo R. 2022. Symphonizing pile-up and full-alignment for deep learning-based long-read variant calling. *Nat Comput Sci* **2**: 797–803. doi:10.1038/s43588-022-00387-x

Received October 2, 2024; accepted in revised form May 2, 2025.