INFO4990 Assignment 1

Sasha Jenner

March 24, 2022

Contents

1 Conferences and Journals

The following list ranks the top conferences and journals relevant to this project.

1. Nature Biotechnology
2. Data Compression Conference (DCC)
3. Bioinformatics
4. IEEE International Symposium on Information Theory (ISIT)
5. Stanford Compression Workshop

According to Scimago, Nature Biotechnology has an h-index of 445 in comparison to DCC, ISIT and Bioinformatics which have 390, 95 and 53 respectively. The h-index is a clear indicator of the impact of these conferences/journals with a higher score meaning more papers having more citations. It is well known that Nature is one of the most prestigious academic journals for a scientist to publish in. However, the DCC is very relevant to this project's research area.

Bioinformatics on the other hand clearly has a great impact and is the journal to publish in apart from Nature Biotechnology in the area of bioinformatics. Furthermore, ISIT is focussed on information theory rather than biology and seems like a great conference with its 2021 edition having been organised to run in Melbourne. Finally, the Stanford Compression Workshop seems to be a very small bi-annual event which is highly relevant to the research topic of genetic data compression.

2 Research Groups

Below is a list of the main research groups working on this research topic. As is expected for an interdisciplinary research topic, there seems to be few groups working on this exact problem. However, various aspects of it seem to be addressed by the following groups.

- Stanford Compression Forum (SCF). In particular: Shubham Chandak and Kedar Tatwawadi who have explored lossy compression of nanopore raw signal data.
- Daniel Lemire and his colleagues at the University of Quebec who have explored integer compression performance optimisation.
- Chen Qianhao from the Institute of Biomedical Engineering in Zhejiang University and his colleagues who have explored lossless compression of sensor signals.

3 Exemplary Papers

1. "Impact of lossy compression of nanopore raw signal data on basecalling and consensus accuracy" by Shubham Chandak et al.

   This paper is exemplary in my opinion since it gives a detailed overview of past research in the area. It presents a significant amount of results on the impact of lossy compression for downstream analysis accuracy of nanopore raw signal data. It is the closest related research paper I have found to the research topic of interest. Furthermore, the authors have structured the paper very nicely into Introduction; Background; Experiments; Results and Discussion; and Conclusion and Future Work.

2. "Decoding billions of integers per second through vectorization" by Daniel Lemire and Leonid Boytsov.

   I also believe this paper is exemplary since it gives a remarkable review of past integer encoding schemes which I would also like to follow in some respects. The writing style is very clear and easy to follow despite the concepts being far from easy to understand. The paper is 29 pages long but the language is clear and concise to read. The authors also include examples following some theory which makes the paper much more readable. Furthermore, the paper's results section is quite exhaustive with many graphs and tables presents many experiments in a clear manner.

4 Research Problems

1. Design a more space and time efficient lossless compressor of nanopore raw signal data.
2. Design a lossy compressor of nanopore raw signal data with significant compression benefits and few analysis drawbacks.

5 Annotated Bibliography

5.1 Research Question 1

[1] Qianhao Chen, Wenqi Wu, and Wei Luo. Lossless compression of sensor signals using an untrained multi-channel recurrent neural predictor. *Applied Sciences*, 11(21), 2021.

This paper presents a new lossless compressor dedicated to sensor signal data called MCDRC. It is built upon a recurrent neural network architecture known as a multi-channel recurrent unit (MCRU) and does not require pre-training. The data it is designed to compress is of great similarity to nanopore raw signal data. The paper compares

MCDRC to gzip, BSC, PAQ and CMIX on four different datasets. The results are fairly promising, especially on more complex data with fewer patterns where it outperforms all other compressors. However, for signal data with simple patterns, such as energy measurements from a typical office environment, CMIX outperforms MCDRC. On the whole, the ideas in this paper present an interesting avenue to pursue when designing a more efficient lossless compressor of nanopore signal data.

[2] Daniel Lemire and Leonid Boytsov. Decoding billions of integers per second through vectorization. *Software: Practice and Experience*, 45(1):1–29, 2015.

This paper introduces new efficient integer encoding schemes SIMD-BP128⋆ and SIMD-FastPFOR which exploit vectorisation on modern processors to reduce costs associated with (de)compression. It gives a detailed review of previous integer encoding schemes including Golomb coding, Rice coding, Elias gamma and delta coding. As well as more modern techniques including Simple family, binary packing and patched coding. The paper also describes the variable byte encoding scheme which is closely related to VBZ: the current state of the art lossless compression technique for raw nanopore signal data. All the presented schemes are designed for a general array of 32-bit unsigned integers, with no particular focus on the nature of the data. These techniques should be considered in parallel with other studies which exploit the signal nature of their data. Nonetheless, this paper delves into architecture-specific optimisations, like exploiting SIMD operations, which might be of interest further along in the project when considering time optimisations.

[3] Scott Gigante. Picopore: A tool for reducing the storage size of oxford nanopore technologies datasets without loss of functionality. *F1000Research*, 6, 2017.

This paper presents Picopore, a Python2 tool for reducing the size of FAST5 files. Unfortunately, the paper presents nothing of interest except for marking an attempt at compressing nanopore data. It mostly reduces the size of FAST5 files by removing redundancies in the storage format or using gzip at level 9, rather than attempting any novel compression techniques on the signal data. It is relevant to the research problem however, since it highlights methodology which has already been conducted towards its solution.

[4] Mikel Hernaez, Dmitri Pavlichin, Tsachy Weissman, and Idoia Ochoa. Genomic data compression. *Annual Review of Biomedical Data Science*, 2(1):19–37, 2019.

This paper gives an overview of the data compression methods used in next-generation sequencing (NGS) genomics. It focuses on the FASTQ, SAM and VCF file formats, which are found further down in the typical analysis pipeline than FAST5/SLOW5. The paper stresses the necessity for effective compressors tailored to genomic data in order to combat the cost of acquiring and maintaining massive amounts of genomic data. It highlights HARC and ORCOM as the state of the art FASTQ read compression strategies, improved upon by SPRING and FaStore respectively as full FASTQ file compressors. For the SAM file format, CRAMv3 (also known as Scrabble) and DeeZ are given as the noteworthy compressors. Finally, TGC and its extension GTRAC are noted as the state of the art compressors for the VCF file format. The authors leave the reader to discover the details about the algorithms of these compression strategies elsewhere. It is relevant as it shows what research has already been attempted in the non-signal space of genomic data.

[5] Divon Lan, Ray Tobler, Yassine Souilmi, and Bastien Llamas. Genozip: A universal extensible genomic data compressor. *Bioinformatics*, 37(16):2225–2230, 2021.

This paper presents Genozip, a genomic data compressor designed to be used on common genomic data formats bar FAST5. It presents impressive compression results, even for files which have are already been compressed. It details its accompanying command line tool

and central algorithm. The algorithm works by dividing the input file into blocks known as *vblocks* comprising of a certain number of lines from the file. Each *vblock* undergoes segmentation followed by compression. Segmentation divides each line into individual data components which are each stored in an amalgamated data structure described as a *context* which stores instructions for the compressor to follow. Compression is then performed on each *context* buffer within a *vblock* using various different codecs depending on the format or options provided. Genozip does not seem to provide a novel compression algorithm, but rather a generic segmenter of genomic data to which other well known compression strategies are employed. Furthermore, although it does not discuss the compression of nanopore signal data of FAST5 files, it does present another attempt at an efficient genomic data compressor which is useful in solving the research problems.

5.2 Research Question 2

[6] Shubham Chandak, Kedar Tatwawadi, Srivatsan Sridhar, and Tsachy Weissman. Impact of lossy compression of nanopore raw signal data on basecalling and consensus accuracy. *Bioinformatics*, 36(22-23):5313–5321, 2020.

This paper evaluates the impact of two lossy time-series compressors (LFZip and SZ) for nanopore raw signal data on basecalling, consensus and methylation calling accuracy. It presents an extensive series of results, documenting the trade off between information loss and downstream analysis accuracy for four different datasets. Both of the lossy compressors used employ the maximum absolute deviation between each original and reconstructed data point as their distortion metric. Given a fixed maximum error rate, LFZip provides slightly better compression than SZ. The paper describes that LFZip differs from SZ by performing uniform scalar quantisation rather than a curve fitting approach before employing entropy encoding. Its results show that after reducing the lossless compressed size by 35-50% (comparing to VBZ), basecalling and consensus accuracy are reduced by less than 0.2% and 0.002% respectively. These results are quite impressive and give confidence to an exploration of lossy compression in this project. They suggest that lossy compression does not lead to drastic structural changes in the read, but rather small disturbances which do not significantly affect downstream analysis.

[7] Shubham Chandak, Kedar Tatwawadi, Chengtao Wen, Lingyun Wang, Juan Aparicio Ojea, and Tsachy Weissman. Lfzip: Lossy compression of multivariate floating-point time series data via improved prediction. In *2020 Data Compression Conference Proceedings (DCC)*, volume 2020-March, pages 342–351, 2020.

This paper presents LFZip, a lossy compressor of multivariate floating-point time series data based on the prediction-quantisation-entropy encoder framework. It uses the maximum absolute error as its distortion function. The encoder first predicts the symbol at time t using the previous reconstructions (i.e. $P(\hat{x}_1, \ldots, \hat{x}_{t-1}) = y_t$). The paper describes two methods for doing this, namely the Normalised Least Mean Square (NLMS) and neural network based predictors. The error $\Delta_t = x_t - y_t$ is then 16-bit quantised with a step of size 2ϵ such that $|\hat{\Delta}_t - \Delta_t| \leq \epsilon$. The final entropy coder step involves applying the lossless compression method BSC to the quantised time series of the differences $\hat{\Delta}_1, \ldots, \hat{\Delta}_n$. The same steps are applied in reverse to decode the compressed stream. The paper presents unremarkable compression results for nanopore raw signal data when using NLMS but outperforms the state of the art, SZ, when using a neural network based predictor. It points to several other lossy compression methods in the literature; Swing door and Critical Aperture are said to retain a subset of the data points; SZ, ISABELA and NUMARCK are most similar to LFZip and use polynomial or regression models to predict to next point followed by quantisation. The paper also notes an efficient lossless

compressor of multivariate floating-point data known as FPZIP which seems to be the inspiration for naming LFZip.