From Bases to Bits
An Evolution of DNA Compression Algorithms

Jared Arroyo Ruiz ('26), Gavin Saxer ('26), Ryan Son ('26) – *Advised by Layla Oesper*
Carleton College, Computer Science Department

Introduction

Since the beginning of the 21st century, **the cost of sequencing genomic data has vastly decreased**. Previously, sequencing a human genome cost an overwhelming $30 million [1]. Today, thanks to technological advancements, **an entire genome can be sequenced for around $80**—a 3,750,000-fold decrease in cost [2]. Due to the reduced financial burden of modern sequencing, economic resources no longer impede the field of genomic sequencing. Yet, the data created within ambitious sequencing projects, such as the 1000 genomes project, has caused what we can only describe as a **"data flood"** [3]. As such, the utilization of **specialized compression algorithms** to handle such extensive amounts of data has proved to be **pivotal in continuing the advancing development of personalized medicine**.

A revolution in genomic data compression came in **2009** with Christley et al's **DNAZip** [5]. Rather than attempting to compress the full human genomic sequence, Christley's group exploited the **similarities between genomes**, only encoding the differences between the **target genome and a reference genome**, creating what we know today as **reference-based compression algorithms**. That is, given that humans share nearly 99.9% of their genetic makeup, Christley's group focused only on the variation across the remaining 0.1%, immensely shrinking the size of data needed to be stored.

Objectives

- Determine flaws within the architecture of DNAZip and establish ways in which restructuring the algorithm may provide additional benefits.
- Compare and contrast non-specialized (Huffman), reference-based (DNAZip and VCF), and non-reference-based (Biocompress 1) compression methods.
- Identify a cost-effectiveness threshold for DNAZip, with respect to non-reference-based techniques.

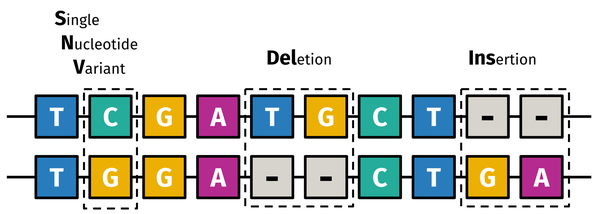
Dataset

Genomic data was gathered from the NCBI genome database. We utilized GRCh38, the current version of the human genome reference assembly, as our reference. To generalize our algorithm, we selected genomic files Ash1_v2.2 and PAN027_mat_v1.0 as they vary in coverage and assembly methods.

Materials & Methods

DNAZip is a reference-based compression algorithm

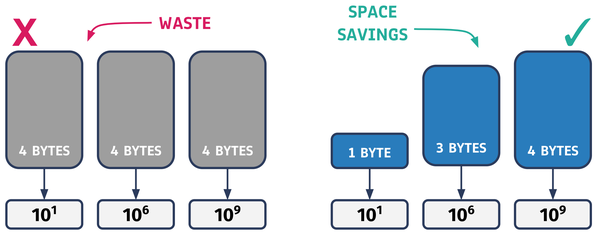
- Reference-based compression algorithms utilize a reference sequence and only encode differences with respect to the sequence. Variations can be broken down into three categories:



VARIANT	ENCODING	EXAMPLE
SNV	<position, nucleotide>	(2, G)
Del	<position, n deletions>	(5, 2)
Ins	<position, sequence>	(8, GA)

Variable Length Integers (VINTs)

- Rather than encoding each position value with a set number of bytes, VINTs allow us to only utilize the necessary amount of bytes to represent a position or integer.



Relative Positions (DELTA)

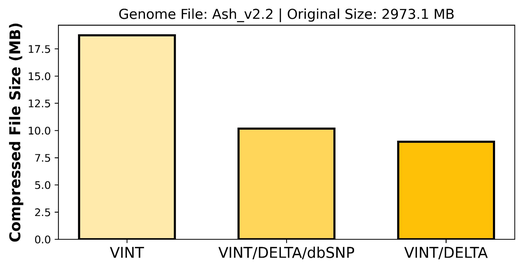
- Space savings are increased by encoding variant positions with respect to the distance from the previous variant.
- To calculate relative positions, subtract the previous variant's absolute position from the current variant's absolute position.

Single Nucleotide Polymorphism Database (dbSNP)

- dbSNP contains common human nucleotide variations which have been observed in sequencing data.
- To reduce SNV encoding size, DNAZip checks if a variant is in dbSNP. If so, it can be represented with a single bit.

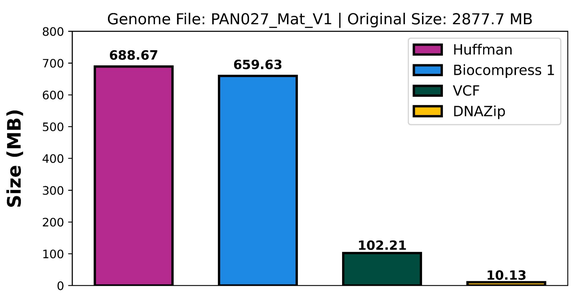
Results

Compressed File Sizes for DNAZip Variants



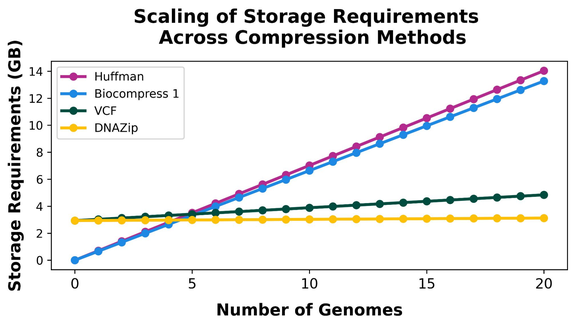
Of the three DNAZip variants we tested, one clear winner emerged in terms of pure file size. The configuration that omitted the use of the dbSNP database, while utilizing both VINTs and DELTA position information, consistently produced the smallest compressed files.

Compressed File Size for Various Methods



DNAZip ultimately beats other methods, though it has an upfront "cost" of including a reference genome. However, this is a one-time fee. Its superior compression ratio pays for the reference after about five genomes, delivering compounding savings from then on.

Scaling of Storage Requirements Across Compression Methods



Conclusions

- The use of **DELTA positions** is vital for DNAZip as it nearly **halves the final compressed file size**.
- While the original DNAZip algorithm was implemented with the use of **dbSNP**, given that the size of the database has increased, utilizing the database is **no longer optimal**.
- Compared to other compression methods (Huffman, Biocompress 1, and VCF), **DNAZip provides the most space savings**.
- While **DNAZip requires a reference genome** (~3.5 GB), when storing more than 5 genomic files, the cumulative **space savings offset this initial cost**.

Future Work

The world of DNA compression, specifically reference-based algorithms, continues to progress. In less than 5 years from the publication of DNAZip, the human genome was able to be compressed further [5]. Yet, further work would attempt to compare DNAZip to more modern reference-based compression methods.

While the use of dbSNP has been proven to be outdated, variations of the dataset have been created which include common insertions and deletions. Further exploration would be done to see if applying such datasets could increase space savings.

Acknowledgements

I would like to thank my fellow group mates for the effort they have placed in this project to make it possible, Mike Tie for providing his technical expertise, and Layla Oesper for her guidance, knowledge, and encouragement!

References

[1] S. Deorowicz and S. Grabowski, "Data compression for sequencing data," Algorithms Mol. Biol., vol. 8, no. 1, pp. 1–13, Jan. 2013, doi: 10.1186/1748-7188-8-25.

[2] F. L. Genomics and L. Fletcher, "The $100 Genome: Where's the Limit?," Front Line Genomics. Accessed: Oct. 05, 2025. [Online]. Available: https://frontlinegenomics.com/the-100-genome-wheres-the-limit/

[3] D. Keiger, "The DNA data flood," The Hub, Fall 2013. Accessed: Sep. 29, 2025. [Online]. Available: https://hub.jhu.edu/magazine/2013/fall/dna-data-flood/

[4] S. Christley, Y. Lu, C. Li, and X. Xie, "Human genomes as email attachments," Bioinformatics, vol. 25, no. 2, pp. 274–275, Jan. 2009, doi: 10.1093/bioinformatics/btn582.

[5] Pavlichin DS, Weissman T, Yona G. The human genome contracts again. Bioinformatics. 2013 Sep 1;29(17):2199-202. doi: 10.1093/bioinformatics/btt362. Epub 2013 Jun 22. PMID: 23793748.