# Compressing Genome and Metagenome Sequences

Badhan Das

October 5, 2022

The number of sequencing data from samples, including genome and metagenome sequences, has increased dramatically as research in Computational Biology and Bioinformatics expands. As this count has grown recently, any project working with genome or metagenome sequences requires a large amount of computational space to conduct its experiments. Again, the databases that store these sequences require a massive amount of storage space. Because research in this field will be extensive, it is high time we focused on how to compress these sequence files more efficiently to organize them better for our research purposes.

There are numerous file compression methods that are widely used for a variety of purposes, including zipping raw sequencing data files (fasta/fastq). When a file or a group of files is compressed, the resulting archive takes 50% to 90% less disk space than the original ones. Some common types of file compression are zip, gzip, tar, fz, rar, 7z, etc. To compress fasta/fastq files, typically, fz and gzip are used. Each of these compression methods compresses data using a different algorithm. While each compression algorithm is unique, they all function in the same way. The goal is to replace common patterns with smaller variables in each file to remove redundant data.

While these compressing tools can handle any possible file type to compress, it will be advantageous if a specific file can be compressed more granularly. We are particularly interested in the genome (nucleotide) and metagenome sequences, typically saved as fasta files. This sequence has a unique format; the whole sequence consists of only four letters: A, T(U for RNA), C, and G. This crucial information can help us think more deeply about compressing sequence files at a granular level. In this paper, we propose a new coding system for shortening the genome sequence. Some existing coding techniques can be briefly discussed to understand the proposed one better. The octal number system uses the numbers $0-7$. While converting a binary number to octal, we take 3 bits and replace them with the equivalent octal digit. This can be viewed in such a way that we are reducing the length of the binary string by a factor of 3. It is the same logic for the hexadecimal number system, but instead of 3, it reduces the length of the binary string by a factor of 4.

Here propose a coding system that uses a set, $S$, of any 16 English alphabets skipping the nucleotides A, T, C, and G (for RNA, skipping A, U, C, and G, or for both, T and U can be skipped). Let $N = \{A, T, C, G, U\}$ be the set of nucleotides. For each nucleotide in the sequence, there can be 4 options: A, T(U), C, and G. So, for two adjacent nucleotides, there will be 16 possible permutations: $AA$, $AT$, $AC$, $AG$, $TA$, $TT$, $TC$, $TG$, $CA$, $CT$, $CC$, $CG$, $GA$, $GT$, $GC$, and $GG$. These 16 permutations can be encoded to each element of $S$. For clarity, let us consider $S = \{B, D, E, F, H, I, J, K, L, M, N, O, P, Q, R, S\}$. Due to this encoding, the sequence size will be reduced by 2. Although it is very straightforward for a sequence having an even length, for an odd length sequence, some other steps need to be done while encoding. For an odd-length sequence, if the encoding starts from the leftmost nucleotide of a sequence, then the rightmost one will be left alone. This alphabet can be left alone and will not create confusion since $S$ does not include any elements of $N$. So, while decoding, an alphabet other than the elements of $S$, or to be more specific, an element of $N$, definitely will mean that the given sequence is of odd length, and the alphabet will remain as it is. This encoded version can be further compressed using compression methods, resulting in a space-efficient compressed file.

While existing compression methods work for most file types, understanding the components of fasta files allows them to be compressed even further. The encoding and decoding algorithms can be implemented on some fasta files to see if they empirically follow this theory.