DNA Compression

Roger Pujol
Universitat Politècnica de Catalunya (UPC)
Barcelona, Spain
roger.pujol.torramorell@est.fib.upc.edu

***Abstract*—In this project I am going to analyse the algorithm described in the paper "DNA Compression Using Hash Based Data Structure" [1].**

The code of this project is Open Source and can be found in: https://github.com/rogerpt32/dna_compression [2].

I. Summary

Nowadays, biologists are producing huge volumes of DNA sequences that makes the genome sequence database grow exponentially. Since the amount of data is getting so big, there is the need to find efficient algorithms to compress it. The regular text compression algorithms doesn't perform very well with DNA sequences. Most of the current DNA compression algorithms exploit the repetitive nature of the DNA sequences, but the algorithm presented here is independent to that.

The algorithm to compress basically creates a hash table with all the possible combinations of the for letters (A, C, G and T) as hashkeys, where each hashkey will be represented by only one byte. After that the algorithm scans each group of 4 letters and store the byte given by the hash of it.

The algorithm to decompress, for each byte of the compressed sequence it searches the hashkey associated to it, which will be 4 letters long.

II. Opinion

I am not an expert in DNA compression data structures, but I would be surprised if nobody before thought about changing the encoding of the letters to a reduced one. In this paper they claim that their compression has exactly a 75% rate, but this is assuming that each letter takes 1 byte, which for a total of 4 different letters is obviously a bad encoding. As even they state, since there are only 4 different letters, they can be described with only 2 bits: A (00), C (01), G (10), T (11). Then, why the algorithm needs a hash table in the first place? They could simply change the encoding to the 2 bit one and get exactly the same final size.

In summary, if this 75% "compression" wasn't done before, this paper could be relevant, but with the simplicity of it I doubt there wasn't any encoding before that had achieved the same level of compression using a very similar method.

III. Experiments

A. Design

The design of a set of experiments to validate the main aspects of the data structure(s) (or the proof of the main theorems in case of a theoretic paper, demo) (15%)

B. Analysis

A critical analysis of the results of the experiments (theoretical results) (15%)

IV. Conclusions

Your personal conclusions (10%)

References

[1] Ateet Mehta and Bankim Patel. DNA compression using hash based data structure. 2:383–386, 2010.

[2] Roger Pujol. DNA compression. https://github.com/rogerpt32/dna_compression, 2019.