Report1 for DNA Storage

# Introduction

# GC-Content and Run-Length Constraint

1 Method Overview

Not to make things too complex at the very beginning, it is reasonable to first consider only GC-content and run-length constraints since these two are mentioned and considered by most other studies.

Wentu Song et al. have come up with a method that can (only) satisfy these two constraints and can theoretically reach the highest code rate of $\frac{2n-1}{2n}$ [Ref]. The main idea of this method is staright forward. It simply tries to enumerate elements in Z_4^n that satisfies run-length constraint with as less as possiable GC-content distance. GC-content distance is defined as the absolute value of the difference between GC-content and 0.5. By enumerating in this way, it is expected that at least 2^{2n-1} elements can be found such that all the elements in Z_2^{2n-1} can be mapped to a subset of Z_4^n, which actually generates an encoding table.

It is shown in [Ref] that if $3 \leq n \leq 35$, at least 2^{2n-1} elements that satisfy run-length constraint (with preceding three nucleotides) can be found in Z_4^n. And according to the demonstration in [Ref], there is no element violating GC-content constraint for $n = 8$ and $n = 10$.

2 Encoding Details

Since n should satisfy $3 \leq n \leq 35$. Hence, the encoding scheme must encode the original data into many short segments which are no longer than 35. Recap that DNA synthesis procedure can generate oligos of 200 nt long. So the run-length validity of an encoded segment should be verified together with preceding three nucleotides and each of the 64 possibilities of preceding three nucleotides will independently have an encoding function.

To generate the encoding table, i.e., the encoding function, the first step is to find a set, say S, of all elements that satisfy run-length constraint in Z_4^n. The second step is to find a subset of S, say G, containing 2^{2n-1} elements with least GC-content distance. The decoding table can be generated by reversing the encoding table.

To encode a message, we first use arbitrary one of the 64 encoding table to encode the first segment. Then for each segment to encode, we choose the encoding funtion according to the previous three encoded nucleotides and use it to encode this segment.

To decode a message, we use the same function for encoding the first segment to decode the first encoded segment. The remaining procedure is similar to that of encoding.

3 Implementation and Test

To find the 2^{2n-1} elements with least GC-content distance, a priority queue can be used to store all the elements that satisfy run-length constraint and then desired 2^{2n-1} elements can be directly extracted from the priority queue.

To store the reversed table, i.e., the decoding function, a hashmap and binary search are both availiable. For binary search, the time comlexity is $O(\log_2(2^{2n-1})) = O(n)$ which is actually $O(1)$ since $3 \leq n \leq 35$.

Hence, the expected time comlexity for encoding and decoding m segments are both $O(m)$. The most time consuming operation is constructing the encoding table since at least 2^{2n-1} elements need to be checked, which leads to a complexity of $O(2^n)$ for both time and space. However, the good news is that, the table need only to be constructed once.

TEST part still in progress......

# LT Code

In progress...

# Future Plan

[1 LT Code]

2 Further Test

3 Improve Performance