Challenge 1

DNA Fountain

Background: You are a ~~spy~~ graduate student in Adv Syn Bio course at UW. A text string was generated and converted using a binary encoding scheme. Blocks of information were combined into 285 total droplets using a Luby Transform (LT) then synthesized as DNA. Figure 1 shows the relationship between blocks of information and the LT droplets. A list of what blocks of information are encoded in each droplet can be found in the luby_blocks.csv file. There is an overall redundancy in information – about 56 blocks of information are encoded in various forms in the 285 droplets. Information redundancy is what makes DNA fountain robust, meaning we do not need to decode all 285 droplets to recover our message. In fact, our message can be decoded in a minimum of n_blocks+1 number of droplets. We are therefore operating at about 5x redundancy.

Why is robustness important? Some sequences of DNA might be difficult to sequence or synthesize. The Luby Transform ensures that any block of information can be stored in a pseudo-random set of letters, and therefore can avoid pitfalls of strict 1:1 encoding schemes. Next generation sequencing was used to sequence the droplets of DNA. You can find the sequencing results in droplet_sequences.fasta.

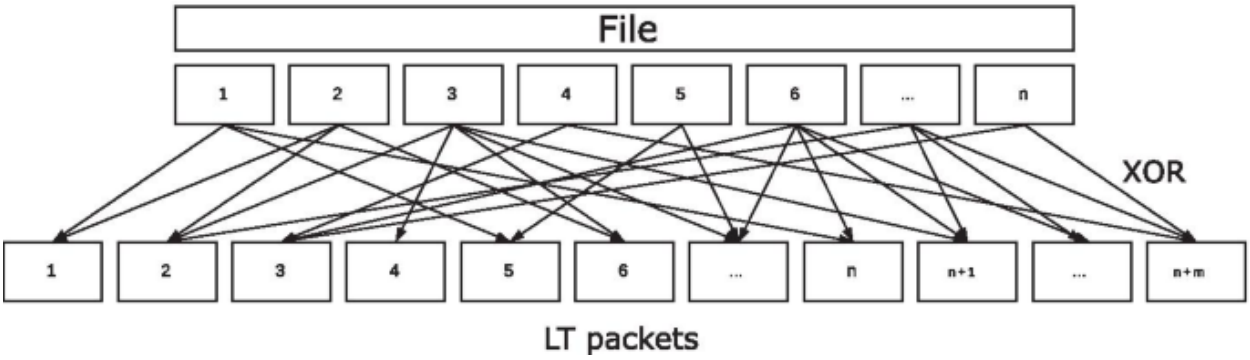

***Figure 1**. Overview of how our original message (file) was split into blocks of information, then combined in various ways using Luby Transform.*

Goal: Decode the DNA droplet information back to the original message. You will need to reverse the Luby Transform using information provided to recover the messages of each block, then stitch blocks of information back together into the correct order to retrieve the binary transform of your original message.

Encoding scheme:

Base	Binary
A	00
G	01
C	10
T	11

Droplet architecture (285 droplets generated):

- [Droplet] = 288 bits
  - [Luby_Index][Droplet_Message][Error_Correction_Code]
    - [Luby_Index] = 16 bits
    - [Droplet_Message] = 256 bits
    - [Error_Correction_Code] = 16 bits

Sequencing headers: Sequencing headers have the following format. The droplet number can be used to find what blocks of information are encoded in the droplet using luby_blocks.csv. Number of degrees is an abbreviation for the number of blocks of information encoded in that given droplet, such that: ***blocks of information in a droplet*** = **number_of_degrees + 1**. This number should match what is encoded in luby_blocks.csv.

Format for header:
>droplet_n[droplet_number]_d[*blocks of information in a droplet*]

Example: >droplet_n273_d2 is droplet number 273, which contains 1 degrees (2 blocks of information from the original file)

Note: You are using illumina sequencing with their 150 nt MiSeq kit for this work. You expect sequencing errors to be very low (1 in 10,000) and therefore do not anticipate needing to use the error correction code in your droplets. Error correction code is present but does not need to be used.

Deliverable: To receive credit, you will need to provide proof that you did this work (and not copied the answer from another individual). The format of this proof can be a writeup of how you went about solving this problem alongside code you generated.