A DNA Memory Translator for Multiple Languages

Yi-Man Huang, Shien-Fong Lin

Abstract - **As the demand for data storage is growing exponentially, higher density and durable storage solutions are necessary. DNA has many potential advantages as a medium for information storage because it is extremely dense, high capacity, low-maintenance and durability. Previous research has focused on ASCII code files or pictures, which ignored expression with other languages encoded with Unicode. We provide a DNA translator for multiple languages and also discuss the influence of entropy in different dataset. The effect of Huffman code encoding and ternary code encoding for different dataset are also compared. Our results indicate that the DNA memory translator system is very efficient for the dataset with low entropy.**

Keywords - **DNA memory; Encoding; Unicode text; Entropy**

I. INTRODUCTION

Data explosion is a phenomenon which describe rapidly increased demand for data storage and is one of the top unsolved issue in modern digital world. According to a report from IBM in 2013, human created 2.5 quintillion bytes of data every day. Furthermore, 90 percent of the stored data in the present world were created in the last two years. Most of the world's data today is stored on hard device made of magnetic and optical media. Despite the improvement of hard drives, the storage ability of hard drives could not catch up with the ever increasing demand for data storage. For instance, huge physical space and poor durability are crucial weakness for hard drives. Finding storage system with significant advances in storage density and durability is an important issue.

DNA is an attractive medium for data storage. This approach is extremely dense, high capacity, low-maintenance and durable. Storing digital data on DNA have great potential [1-4]. It overcomes all of the shortcomings of current data storage medium. For instance, the DNA-based storage will not break down when exposed to magnetic fields or to extreme temperatures. It is long-lasting because it is not susceptible to data loss in the event of power loss. The half life of DNA is over 500 years in harsh environments [5]. As long as the medium is kept in a relatively stable state, DNA based archival storage system could last thousands of years without degradation.

*This study was supported in part by the Ministry of Science and Technology, Taiwan, R.O.C. under the grant number MOST 106-2221-E-009-061

Yi-Man Huang and Shien-Fong Lin are with the Institute of Biomedical Engineering, College of Electrical and Computer Engineering, National Chiao Tung University, Hsinchu, Taiwan (e-mail: linsf5402@nctu.edu.tw).

There is a rapid progress in DNA storage since 1999. In that year, Clelland et al. encoded and recovered a message containing only 23 characters [1]. In 2013, Goldman et al. made a more scaling improvement, they successfully recovered a 739 kB message [3]. In 2015, Grass et al. recovered a 83kB message without error. They used a Reed-Solomon code [6] and their dataset needed to be synthesized over 5000 DNA strands [7]. In 2016, James et al. demonstrated feasibility and random access of the proposed encoding with wet lab experiments involving 151 kB of synthesized DNA [4]. Although the volume of data that can be synthesized today is limited mostly by the cost of synthesis and sequencing, the growth of biotechnology industry is predicted. Consequently, the magnitude of cost reductions and efficiency improvements can be expected.

This paper presents an architecture for a DNA translator system for data expressed in multiple languages. The basic structure of our system is based on the structure used by James et al [4]. Existing approaches have focused on ASCII code files or pictures, which ignored other languages data expressed in Unicode. In our work, the new system offers a DNA translator providing multi language encoding. The influence of entropy in different dataset also discussed according to our results. Our results demonstrate that the system is very efficient for the dataset with low entropy. We also provide a pair of primers which is suitable for the DNA archival storage system. We use simulations to support the feasibility and efficiency of our system design.

II. THE DNA STORAGE SYSTEM

A DNA storage system consists of a DNA synthesizer, DNA pools, polymerase chain reaction (PCR) thermocycler and a DNA sequencer [4]. Due to the limitation of DNA synthesis technology, the length of DNA strand is only 100 to 200 nucleotides long. Therefore, the data supposed to store is necessary to cut into small data blocks and each of data blocks needs their own address to aid the recomposed of the data.

Digital data can be converted into the DNA form and then use the DNA synthesizer to write the data into DNA strands. After DNA synthesizing, the DNA strands are stores in the DNA pools which is a storage container with compartments that store pools of DNA that map to a volume. Then, the specific strands are amplified using the PCR process. Next, use the DNA sequencer to read the DNA sequences and convert them back into digital data.

III. THE COMPOSITION OF DNA SEQUENCE

Due to the technical restrictions of recently DNA synthesis technology, the length of DNA sequences is limited. Data more than the hundreds bits cannot be synthesized as a single strand of DNA. Therefore, data need to be split into small

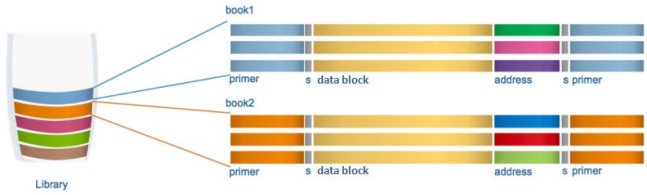


Figure 1. The composition of DNA sequences

block and we also need to mark the order of each strand by address. In addition, DNA pools contains data for many different sources which are unrelated to a single read operation. To overcome these two challenges, we use the same pattern of DNA sequence composition as James et al. [4], as shown in Figure 1. Each sequence is composed of four parts: primers, payload, data block and address. Below we describe every parts of the sequences in detail.

Payload

Data desired to be stored is broken into small data blocks. Aiding for the decoding, two sense nucleotides indicate if the strand has been reverse complemented which is done to avoid certain pathological cases [4]. The sense nucleotides are shown as "S" in Figure 1.

Primers

We attach the primer sequences to each end of the strand. These primers sequences are used for the PCR process, and let the PCR to have a high throughput which can selectively amplify with the chosen primer. To achieve random access in DNA data storage system, effective primers are needed. We also knew that there are some typical restrictions for primers in PCR process and sequencing. We developed an algorithm to produced appropriate primers, and will discuss the primer design algorithm in detail at primer design section.

Address

Each DNA sequence will have an address involved, and it will let us easily identify the order in all the data blocks. The length of address is fixed, and it will be attach to the end of the encoded data block.

Data block

Because of the DNA synthesis technical restrictions, data which we supposed to store will be cut into small data blocks. In order to storage the data using the DNA format, we also need to convert the digital data to the DNA nucleotide form. There are many ways to translate the digital data into the DNA format, we will explain the encoding methods in detail in next section.

IV. DNA Translator System

Our DNA Translator System provide users to translate the txt file to DNA format and the given file which is encoded by Unicode. Figure 2 show the process of our DNA translator system. First, we read the txt file in our system and decide how many pairs of primers are needed. Next, we use our primer design algorithm to produce suitable numbers of primers. Then we encode the data and break them into data blocks in specific length. There are two major data encoding ways in our

Figure 2. Process of DNA translator system.

Figure 3. Design of Huffman encoding programming.

system: Ternary code and Huffman code [8]. After data encoding, we attach the address, payload and primers to all the data blocks. We save the encoded sequences and return the file to the users in the last place. Below we discuss each process of our system in detail.

A. Primers Design

At write time, primers are added to each end of the strands. At read time, those primers are used in PCR to amplified the desired strands. Appropriate primers not only offer data random access ability but also lower the error rate in PCR. However, there are many restrictions and details need to be paid attention in primers design. To provide effective primers pairs, we design an algorithm which automatically produce a suitable and unique primers pair for the input data.

First, we produce two random primers which length of it is in a specific range. Then we calculated the primer melting temperature (Tm) and make sure the Tm value is in the range of 52 to 58 degrees Celsius (ºC). We also assure the differences of Tm value of these two primers is less than 1 ºC. We also eliminated the primers if any of the following is true: (1) The GC content is outside the specified range, (2) More than 3 G's or C's in the last 5 bases at the 3' end of the primer, (3) Presence of the primer secondary structures, including hairpins, self dimer and cross dimer, (4) Repetitions occur, (5) Unstable 3' end. After using the algorithm which we developed, we will get a pair of primer and they are appropriate for the following procedures.

B. Data Encoding Methods

Recently, most English texts use ASCII code to encode the symbol in the digital data. However, English is not the only character used in the world. For extension to encoding other languages, Unicode showed up. Unicode is developed by the Unicode Consortium, which defines a set of letters, numbers, and symbols that represent almost all of the written languages in the world. Its success in unifying character sets has led to widespread use in the creation of software. Compare to ASCII code, Unicode uses 16 to 32 bits to show each symbol, ASCII code only uses 8 bits.

Our translator system supports converting the Unicode encoded file to the DNA format which means that we can translate almost all of the written languages in the world.

Representing Data in DNA

DNA consists of four types of nucleotides: adenine (A), cytosine (C), guanine (G), and thymine (T). The obvious

approach to store binary data in DNA is to encode the binary data in base 4, producing a string of n/2 quaternary digits from a string of n binary bits. The quaternary digits can then be mapped to DNA nucleotides (e.g., mapping 0, 1, 2, 3 to A, C, G, T, respectively). However, DNA synthesis and sequencing processes still has a variety of errors, requiring a more delicate encoding [1].

The Goldman's approach avoids the homopolymers — repetitions problems. The repetitions of the same nucleotide that significantly increase the sequencing errors. Goldman's approach modified the base 4 encoded way by encoding binary data in base 3 Huffman code[3]. The base 3 Huffman code encode each bit of the data with one of the three nucleotides different from the previous one used. Another trait of Huffman code is that if the probability of the symbol appearance is higher the encoded code is shorter. This trait make people choose Huffman's way to decrease the length of the code.

We compare the ternary code and the base 3 Huffman code to encode the data file. Use Chinese characters for example, each symbol needs 11 bp to represent if we choose the ternary code. The base 3 Huffman code will be more effective if the symbol shown in the file repeated more. Differences from the recently research, we do not use the fixed Huffman table for ASCII code. We established the specific Huffman table for each file, which is according to the frequency of symbols appearance in the file. This method let the encoding more flexible and also can shorten the length of code. We will discuss the detail of our data encoding method in next section.

Ternary code and Base 3 Huffman code

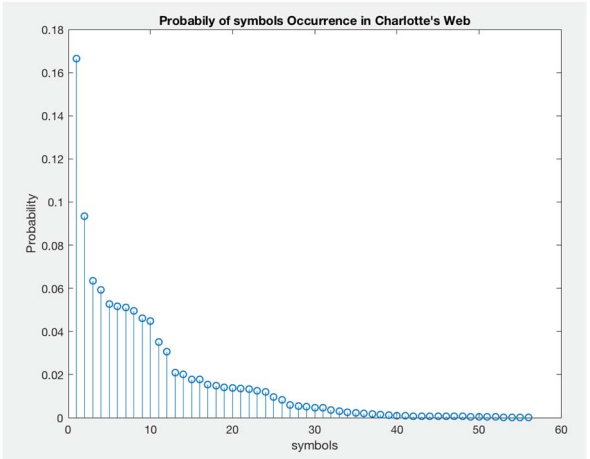
Ternary code encode each bit of the data with one of the three nucleotides different from the previous one used. We calculate the needed length of every symbols in file present in DNA format and chosen the longest length to encode the data. These length is depending on the number of the Unicode. For example, compare to the halfwidth font, the fullwidth fort needed longer length in DNA format.

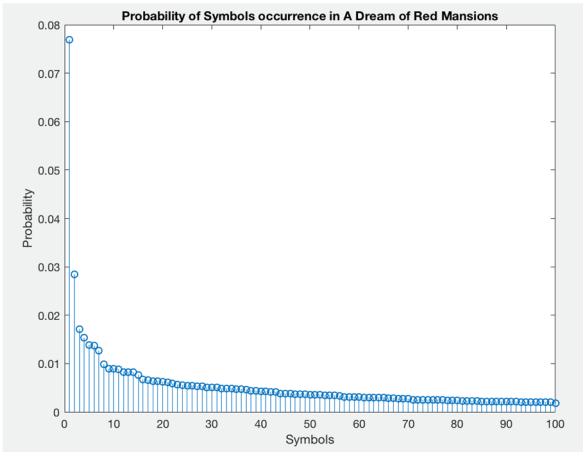
Huffman algorithm derives its own table from the estimated probability or frequency of occurrence for each possible value of the source symbol. More common symbols are generally represented using fewer bits than less common symbols. The output can be viewed as a variable-length code table for encoding a source symbol[8]. We use ternary Huffman code also called base 3 Huffman code. Figure 3 shown the logic of our Huffman encoding programming design.

First, we calculate the appearance frequency of every symbols in the file. Second, we established the Huffman dictionary according to the previous outcomes. This Huffman table is only belonging to this file which means that every table we created is unique. Third, we encode the data using the Huffman dictionary we established previously and broke them into data blocks in specific length. And then, we convert the data blocks into DNA format and each bit of the data will be convert to one of the three nucleotides different from the previous one used. The output is the sequences of the data blocks and they will be compose with other parts of the entire

```
How are you?
TCGTCGCGTGACTCTATGTGTGCATCTGATGATATGA
你好嗎？
TCACAGTCGCGT
안녕하세요？
TCACGAGATATCGACAGTG
```

Figure 4. Base 3 Huffman Code Examples





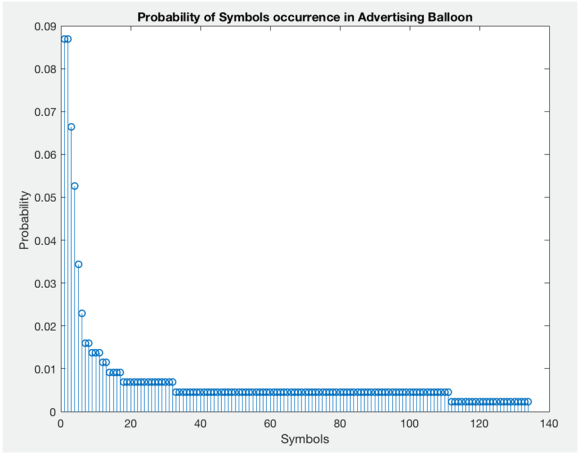


Figure 5. Probability of symbols occurrence

sequences. A few simple Base 3 Huffman code encoding examples are shown in Figure 4. Because the Huffman tree is unique for every files, so this method let the encoding more flexible and also can shorten the length of code. We will discuss the detail of our programming results in next section.

V. SIMULATION

A. Dataset

Our system can read different language files encoded by Unicode and translate them to DNA format. Three dataset are tested in our simulation. The first one is chapter 1 of "Charlotte's Web" in English. The reason we choose it is because "Charlotte's Web" is considered as a classic of children's literature. Also, it is listed as the best-selling children's paperback by Publishers Weekly in 2000. There are 5306 words in chapter 1 and 56 symbols left after eliminating repeat symbols. The second dataset we use chapter 1 of "A Dream of Red Mansions" for shown. "A Dream of Red Mansions" is one of China's Four Great Classical Novels. It was a masterpiece of Chinese literature. There are 7,009 words in chapter 1 of this novel and has 1,226 symbols left after eliminating repeat symbols. The third dataset we use lyrics of the song "Advertising Balloon" by Jay Chou. There are 437 words in the lyrics and 134 symbols left after eliminating repeat symbols. Tables 1 shows the information of these three dataset.

B. Method

There are two encoding method in our simulation, which is Huffman encoding and ternary encoding. For Huffman encoding, the first step is to build Huffman dictionary. The Huffman dictionary is created according to the symbols' frequency of occurrence in the file. In figure 4, we could see the distribution of the symbols' frequency of occurrence in three dataset. In "A Dream of Red Mansions", only the first 100 data was shown and others probability of symbols occurrence is 0.0001. Entropy is a measurement of data average information content. The formula us shown below:

$$H(X) = \sum_{i=1}^{n} P(x_i)I(x_i) = -\sum_{i=1}^{n} P(x_i)log_b P(x_i). \quad (1)$$

The entropy of the three dataset is 1.36, 2.60 and 1.88 in order. Because the establishment of the Huffman dictionary is based on the probability distribution of symbols occurrence. We could expect higher entropy to have longer average length of a symbol present in DNA format.

C. Results

The results of two encoding method applied to the three dataset mentioned in the dataset part are shown in Table 2. The total numbers of DNA sequences and the average length of a symbol present in DNA format are compared for three test dataset. The second dataset, which has highest entropy as shown, has longest average length of a symbol present in DNA format. The result is consistent with our expectation from the information of the entropy.

TABLE I. INFORMATIONS OF DATASET

	Numbers of Words	***Numbers of Symbols***
Charlotte's Web	5306	56
A Dream of Red Mansions	7009	1226
Advertising Balloon	437	134

TABLE II. RESULTS OF TWO ENCODING METHOD

	Ternary code		**Huffman code**	
	# of DNA sequences	***average length of a symbol (bp)***	***# of DNA sequences***	***average length of a symbol(bp)***
Charlotte's Web	307	11.6	111	4.2
A Dream of Red Mansions	495	14.1	276	7.5
Advertising Balloon	31	14.2	13	5.6

VI. DISCUSSION

Compare ternary code and Huffman code, Huffman methods can more effectively increase the capacity of DNA storage as demonstrated. It can reduce 45% of the DNA sequence in second dataset and 65% in first dataset. This result shows that our system can reduce the cost of DNA synthesis. Also, we can found the fact that using Huffman code, the average length of a symbol present in DNA diverge between data sets. Comparing the second and the third data set, which are both Chinese data, third data set only needs 5.6 bp to present one symbol in average while the second data set needs 7.5 bp when using Huffman code. The reason is that Huffman dictionary is built base on the occurrence probability in each dataset. Although the average length to present a symbol in DNA format using Huffman code in third dataset is still longer than the English case, the reducing rate of average length of a symbol present in DNA compare to ternary code is 61% which is near to the English case's 63%. These results could be predictable according to the entropy, so we could say that the influence of entropy is higher than the used language of the data.

From the previous results, we know that language used in dataset is not really important to our system. Even the numbers of words will not affect the average length to present a symbol in DNA format. Entropy and the numbers of un-repeatable symbols are the main reason to affect the efficiency of DNA data storage. Because we create Huffman dictionary for each file, so we will know the entropy of each file and can expect the numbers of sequences we may get after the translation. This results shows that our DNA translator system is providing an effective DNA translator method.

VII. CONCLUSION

Base on the advantage of high density and durability, DNA-based storage has the potential to be the ultimate solution of archival storage. By applying our encoding

method, the DNA-based storage could be implement by more efficient way. However, it is still not practical because the restriction of DNA synthesis and sequencing. Nevertheless, thanks to the advances in biotechnology industry, both the techniques are improving at exponential rate. While silicon technology is coming to a bottleneck, we believe it is time to consider cooperate silicon and biochemical systems for storage system design. From the previous decades, biotechnology has greatly benefit from the computer industry. Maybe it is the right time to borrow back from biotechnology industry and to enhance the ability of computer storage system.

REFERENCES

[1] C. T. Clelland, V. Risca, and C. Bancroft, "Hiding messages in DNA microdots," Nature, vol. 399, no. 6736, p. 533, 1999.

[2] D. G. Gibson et al., "Next-generation digital information storage in DNA," Science, vol. 329, no. 5987, pp. 52-6, Jul 2 2010.

[3] N. Goldman et al., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," Nature, vol. 494, no. 7435, pp. 77-80, Feb 7 2013.

[4] J. Bornholt, R. Lopez, D. M. Carmean, L. Ceze, G. Seelig, and K. Strauss, "A DNA-Based Archival Storage System," pp. 637-649, 2016.

[5] M. E. Allentoft et al., "The half-life of DNA in bone: measuring decay kinetics in 158 dated fossils," Proc Biol Sci, vol. 279, no. 1748, pp. 4724-33, Dec 7 2012.

[6] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," Journal of the society for industrial and applied mathematics, vol. 8, no. 2, pp. 300-304, 1960.

[7] R. N. Grass, R. Heckel, M. Puddu, D. Paunescu, and W. J. Stark, "Robust Chemical Preservation of Digital Information on DNA in Silica with Error-Correcting Codes," Angewandte Chemie International Edition, vol. 54, no. 8, pp. 2552-2555, 2015.

[8] D. Huffman., "A Method for the Construction of Minimum-Redundancy Codes," Proceedings of the IRE, pp. 1098-1101, 1952.