THE MAMMOTH INTERNATIONAL CONTEST ON OMICS SCIENCES: DNA STORAGE TRACK

Yi Lu[1,2,*],Chenghao Li[3,*], Yun Ma[1] and Xin Zhang[4]

[1] Capital Normal University, Beijing, China.
[2] University of Liverpool, Liverpool, United Kingdom.
[3] Chongqing University of Technology, Chongqing, China.
[4] Peking University, Beijing, China.

*Equal contribution

Keywords: DNA Storage.

Abstract. DNA molecules, characterized by high information density, long-term stability, and ease of backup, are considered a potential future data storage medium. DNA storage represents an interdisciplinary research direction combining biotechnology (BT) and information technology (IT). To reduce the cost of information writing and improve data recovery accuracy, it is essential to develop or optimize corresponding algorithms. Focusing on image data within the realm of bioinformatics, efficient and accurate DNA storage positioning based on encoding and decoding schemes must be devised. Additionally, considerations for memory usage and computational speed are crucial in this endeavor.

1 Introduction

DNA molecules, as the primary carriers of genetic information for Earth's life forms over millions of years, have recently been recognized for their enormous potential in the storage of digital information. The question of "Can DNA serve as a digital information storage medium?" is listed by Science as one of the 125 key scientific questions. In comparison to traditional information carriers such as optical discs, magnetic disks, and flash drives, DNA molecules as information carriers offer advantages such as high information density, long-term storage, and low maintenance costs. DNA storage is a research field at the intersection of biotechnology (BT) and information technology (IT).

As one of the foundational core technologies for DNA storage, the bit-to-base encoding and decoding algorithm serves as a bridge between binary information and DNA information. It is designed to enhance the compatibility of DNA sequences obtained through the conversion of binary information with existing supporting technologies (such as DNA synthesis, DNA sequencing, PCR amplification, etc.) while minimizing potential base errors introduced by these supporting technologies. Simultaneously, encoding and decoding algorithms with higher information density contribute to reducing the cost of DNA synthesis, laying the groundwork for industrial applications.

Therefore, in recent years, developing bit-to-base encoding and decoding algorithms for DNA storage that achieve high information density, compatibility, and error correction has significant implications for the underlying support of DNA storage process development and applications.

2 Method

Task Setting. The DNA storage bit-to-base conversion encoding for spatiotemporal volume sliced images aims to achieve the recovery of the original image under a specified error rate using

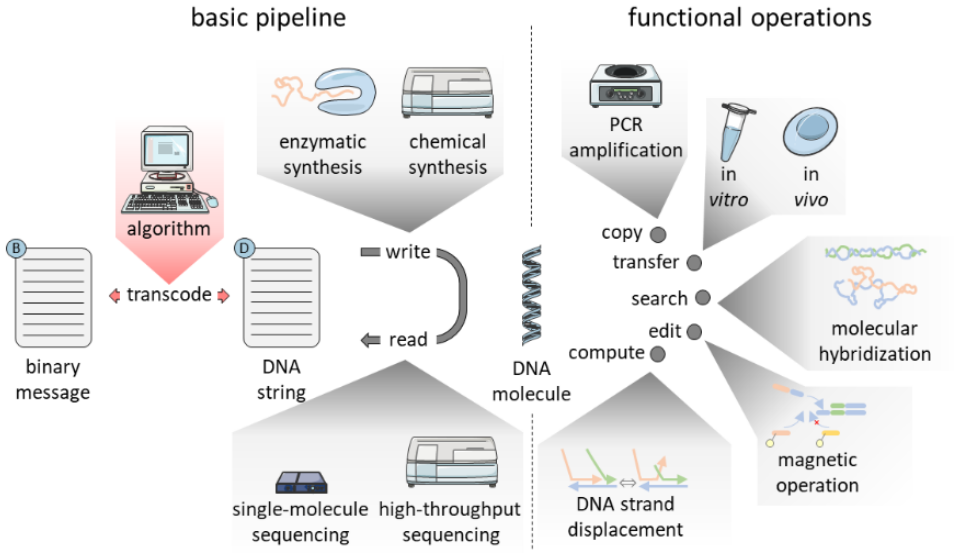

Figure 1: Caption. (This image is taken from [1])

fewer bases. The goal is to enhance compatibility with synthesis and sequencing technologies while ensuring that data recovery closely approximates the original image.

A transcoding task is divided into two parts: encoding (from image to DNA sequence) and decoding (from DNA sequence to image). Through the designed bit-to-base encoding and decoding algorithm in the image_to_dna process, an image is transformed into a list of DNA strings. This string list is then converted into a FASTA format file during the evaluation or competition process. In bioinformatics, the FASTA format is a text format used for recording nucleic acid sequences, commonly represented as follows:

seq_1	seq_2
ACGCTGACGG	TTCAGTCATC

In practical scenarios, this file will be recognized by DNA synthesis devices, and the corresponding DNA molecules will be synthesized accordingly. When information needs to be retrieved from the DNA molecules, DNA sequencing devices will be used to obtain the sequence information of the DNA molecules, and a FASTA format file will be created (the sequence order in this file is not consistent with the sequence order in the FASTA format file used for synthesis). By interpreting this FASTA format file and utilizing the dna_to_image process based on the designed bit-to-base encoding and decoding algorithm, these DNA sequence information will be decoded into an image file.

Evaluation Metrics. 1) The reduction in the total number of bases in the library DNA obtained through conversion, i.e., encoding density, constitutes 20% of the total score. 2) The compatibility of the DNA sequence with synthetic/sequencing technologies, including factors such as the GC content of the sequence, maximum single-base repetition, and distribution, holds a weightage of 30% in the overall score. 3) The introduction of 3% errors in the work, encompassing 1.5% mutations, 0.75% insertions, and 0.75% deletions, and the subsequent decoding to achieve a high image restoration degree measured by the SSIM value, contributes to 50% of the total score. Here, the 3% error rate signifies that the proportion of erroneous bases to the length of the sequence is 3%. [2]

For an image with a width and height, the assumed generated sequence's length during the

design process should be ensured to be not less than 100 and not more than 200.

The evaluation metrics for this task are as follows:

$$0.2\times\min\left(1-\frac{k\times l}{n\times m\times 24},0\right)+0.3\times\left[\frac{1-\frac{\min(h,6)-1}{5}}{2}+\frac{1-\frac{\min(c,0.3)}{0.3}}{2}\right]+0.5\times\left[\frac{\max(s,0.84)-0.84}{0.16}\right]$$

Coding process. Our approach is concise yet direct, aimed at finding a codebook capable of resisting a certain number of substitution, insertion, and deletion errors. We directly encode binary sequences, and decoding involves the inverse operation according to the codebook. In terms of error correction, we have devised a sophisticated Levenshtein function to determine whether a sequence can be transformed into another sequence under specific constraints. Through this assessment, we identify sequences that meet the criteria as correct encodings. In this competition, we have also introduced a step for compressing images. The overall framework is straightforward, with the key emphasis on finding an effective codebook.

The simplicity and error-resistant capability of our approach make it feasible for practical applications, with the **quest for an optimal codebook** being the critical focus.

Computing Resource is All We Need. In our approach, we fixed the codebook and then performed the aforementioned operations. To find such a codebook, we employed a relatively straightforward strategy: under given compatibility constraints, we filtered through $4N$ sequences of length N and identified pairs with an edit distance greater than twice the maximum possible error scenario. We optimized this approach specifically for the given competition to ensure better practical implementation. The optimization primarily focused on two aspects: enhancing the edit distance calculation algorithm and refining the GPU implementation strategy.

Firstly, for a more refined edit distance calculation, we drew inspiration from the Levenshtein distance algorithm, utilizing dynamic programming to compute edit distances with a time complexity of $O(n^2)$. If considering the calculation of edit types (such as modifications, insertions, and deletions), the complexity would increase. However, by only considering error counts within a certain range, the complexity could be reduced to $O(kn)$, where k is a relatively large constant (related to insertions and deletions). Storing the pair of error counts could be efficiently implemented on the GPU. Therefore, we also encoded the error types as binary sequences, subsequently converting them to integers for further evaluation. This modification significantly reduced the overall complexity, especially when considering scenarios with given insertions and deletions.

Secondly, we utilized CUDA for GPU traversal. Recognizing that considering a set of pairwise edit distances greater than n does not require evaluating all possible combinations, we implemented a pruning strategy. If we already had the first code in a pair, we could exclude certain scenarios when searching for the second code, thereby eliminating unnecessary computations. This "top-down" algorithm, as the number of codewords increased, posed challenges to GPU memory. To address this, higher-level CUDA programming skills were necessary. Alternatively, a "bottom-up" approach could be considered. By constructing sequences starting from the first base and utilizing the refined edit distance, we could compare the changes in edit counts for subsequent bases and other sequences. Although this "bottom-up" approach would offer better time and space complexity, we did not implement it due to the competition timeline. Additionally, there is still room for further optimization in the edit distance algorithm.

The aforementioned strategies and optimizations constitute the core of our approach, providing practicality and feasibility for effectively finding a robust codebook. In real-world applications, the search for an optimal codebook remains a crucial research focus.

Two expansion methods. In our research, we have observed that the information capacity of DNA storage is relatively limited, making expansion a crucial task. We have employed two expansion methods, both of which have demonstrated superiority in the current competition environment.

The first expansion method involves appending terminal positions. We have found that by adding additional bases at the end of each sequence to serve as padding, we can replace the role of sequential positions. It is important to note that this method is effective specifically in the context of the current competition. Specifically, we have implemented terminal position padding by adding sequences such as "AGA" to each sequence, enhancing the distinguishability between sequences. Although we cannot determine the exact change in sequence length, in the competition, we only need to ensure that the number of insertions and deletions remains within 1. Thus, this method enables clever expansion while maintaining relatively stable information capacity.

The second expansion method involves codebook rotation. While keeping the codebook consistent, we perform rotations on the bases A, G, C, and T, and filter out a set of codes with unchanged compatibility. As the edit distances remain constant after rotation, this linearly expands the information capacity. Through this operation, we flexibly adjust the codes while maintaining consistency, fully leveraging the potential information capacity of DNA storage.

Both of these expansion methods play a crucial role in our research, providing effective approaches to enhance the information capacity of DNA storage.

Model analysis. In our implementation, we established a Translator class to establish the correspondence between binary units and DNA bases. Subsequently, we designed the Core class to store relevant information for DNA storage, such as the total number of units, sequence positions, and sequence length. Finally, we created the Core_Cluster class to encode different configurations of the Core.

Once we determine the codebook, as well as the number of information units and sequence positions, the storage capacity's upper limit and the required number of bases to reach this limit can be calculated.

Specifically, after determining the codebook, information unit count, and sequence position count, relevant parameters such as the upper limit of storage capacity and the number of bases needed to reach this limit can be calculated. Here, we present the scenario where the information unit count is less than or equal to 5, as larger scenarios may not be practically applicable.

Information Units	Storage Capacity	Bases Required
1	33,554,432 (32.00 M)	2,181,038,080 (2.03 M)
2	16,777,216 (16.00 M)	545,259,520 (520.00 M)
3	6,291,456 (6.00 M)	136,314,880 (130.00 M)
4	2,097,152 (2.00 M)	34,078,720 (32.50 M)
5	655,360 (0.62 M)	8,519,680 (8.12 M)

Table 1: 13 units-130 length-Core related parameters

In the absence of considering density scores, our model is capable of fully storing 381M of data. Here, the 381 refers to the size of **any binary sequence**, whether it represents an image, video, audio, or text. As long as the required storage space is within 381M, we can handle it. However, considering that this competition includes the evaluation of density scores, we decided to use **5 information units** for consideration. As shown in the table above, this encoding method maps 7.44M of binary data to 93.12M bases, which is a ratio of **12.516** (taking into account the occupancy of sequence units). In other words, to achieve a high score in this competition, it is

Information Units	Storage Capacity	Bases Required
1	66,584,576 (63.5 M)	4,165,206,016 (3.88 G)
2	33,292,288 (31.75 M)	1,041,301,504 (993.06 M)
3	12,484,608 (11.91 M)	260,325,376 (248.27 M)
4	4,161,536 (3.97 M)	65,081,344 (62.07 M)
5	1,300,480 (1.24 M)	16,270,336 (15.52 M)

Table 2: 1 Core_Cluster related parameters

Information Units	Storage Capacity	Bases Required
1	381 M	23.28 G
2	190.5 M	5.82 G
3	71.46 M	1.45 G
4	23.82 M	372.42 M
5	7.44 M	93.12 M

Table 3: Rotational Expansion Core_Cluster Parameters

necessary to ensure a compression ratio of **over 12.516** for the images. The characteristic of the images in this competition is mainly reflected in the fact that the majority of the regions are black, with data present in only a small portion. When SSIM decreases by 2-3 percentage points, it essentially achieves a compression effect of over **100 times**. (Therefore, despite the presence of high coding redundancy, the information density involved in the competition is not high.)

3 Results

As our model is only influenced by the information density of the data (and only requires the compressed size + SSIM to calculate the total score), it does not matter what specific data is used. Therefore, let's use the available data to estimate the scores for the upcoming tests: (For convenience of calculation, the number of bases is uniformly calculated at the maximum value, which means the obtained total score is the **lower bound** of the score)

The trap question corresponds to the highest density in our test dataset, which means our model will score no lower than 86.9 points. The average score for all the above scenarios is 92.62481 points (this is only a lower bound estimate). In terms of computational performance, both encoding and decoding can achieve $O(n)$, while error correction takes a bit longer, but our error correction is only triggered when decoding is unsuccessful. So the overall complexity of encoding and decoding is approximately $O(n)$. After testing, the overall encoding and decoding time can be controlled **within 15 minutes**. Additionally, due to the high acceleration ratio in our encoding architecture (as each sequence's encoding and decoding are independent), the overall time for encoding and decoding can be even lower.

4 Discussion and Conclusion

Taking into consideration the requirements of this competition, our work primarily focuses on the rapid calculation of edit distances under specific conditions and the generation of a corresponding set of code tables given error-correction capabilities and encoding compatibility. While some of our optimization strategies were devised to meet the specific demands of this competition, such as compression and padding, the essence of our approach is to provide error-correction-capable encoding.

We observed that many other competitors also need to consider error-correction capabilities when dealing with information storage. Therefore, we believe it is meaningful to offer a set of

Sample	Compressed Size (MB)	Base Count (MB)	Compressed SSIM	Density	Compatibility	Recovery	Total
Trap Question	5.958	93.12	0.97595	87.28	90	84.97	86.94
Test Sample	1.205	93.12	0.99607	99.84	90	97.54	92.93
1	2	93.12	0.99781	99.73	90	98.63	93.47
2	2.19	93.12	0.99754	99.70	90	98.46	93.39
3	2.03	93.12	0.99763	99.72	90	98.52	93.41
4	3.40	93.12	0.99698	99.54	90	98.12	93.21
5	3.41	93.12	0.99681	99.53	90	98.01	93.16
6	3.90	93.12	0.99653	99.47	90	97.83	93.07
7	4.39	93.12	0.99308	99.40	90	95.67	91.99
8	2.54	93.12	0.99743	99.65	90	98.39	93.35
9	2.19	93.12	0.99761	99.70	90	98.50	93.41
10	2.08	93.12	0.99775	99.72	90	98.59	93.45

Table 4: Results for Different Samples

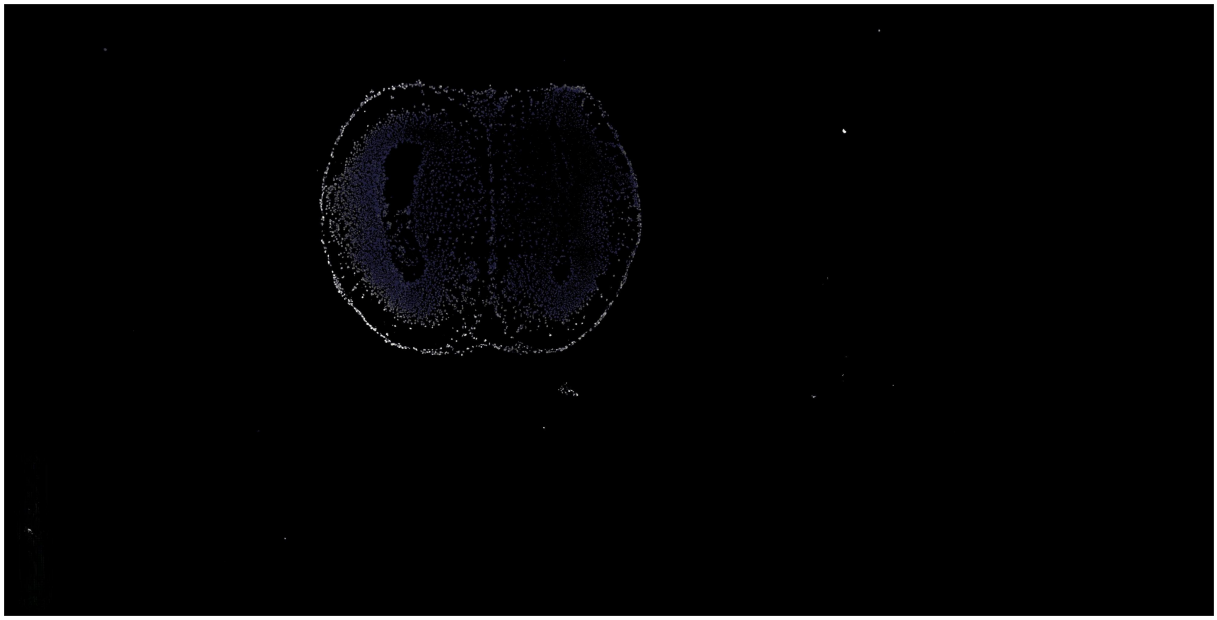

Figure 2: Data sample

encodings with error-correction capabilities. In our initial research, we attempted to search for relevant literature on using VT codes for error correction but found that the compatibility scores of these encodings tend to be lower. Moreover, the ability to correct multiple errors appears to exist primarily at the theoretical level and is challenging to implement in practical algorithms (possibly due to limitations in our technical proficiency).

Building on these observations, we propose a new approach—using multiple code tables within the same sequence, each with different error-correction capabilities. By storing crucial information in tables with higher error-correction abilities and less critical information in those with lower error-correction capabilities, we can enhance the information density of DNA storage. In such cases, we would employ a rotational expansion method to extend the application range of the code tables by at least a factor of 6, thereby more fully utilizing the potential information capacity of DNA storage.

Additionally, we mentioned a ”bottom-up” approach in the process of searching for code tables, wherein sequences are incrementally constructed, leveraging detailed edit distance calculations. If ”finding code tables” itself is a meaningful task, then adopting a ”bottom-up” strategy becomes even more crucial.

These contributions are not limited to this competition alone but also hold significance for scenarios requiring encoding with error-correction capabilities and research directions seeking to enhance information density in DNA storage.

References

[1] Haoling Zhang, Zhaojun Lan, Wenwei Zhang, Xun Xu, Zhi Ping, Yiwei Zhang, and Yue Shen. Spider-web enables stable, repairable, and encryptible algorithms under arbitrary local biochemical constraints in dna-based storage. *ArXiv*, abs/2204.02855, 2022.

[2] Lifu Song, Feng Geng, Zi-Yi Gong, Xin Chen, Jijun Tang, Chunye Gong, Libang Zhou, Rui Xia, Ming-Zhe Han, Jing-Yi Xu, et al. Robust data storage in dna by de bruijn graph-based de novo strand assembly. *Nature Communications*, 13(1):5361, 2022.