

Storing Images in DNA via base128 Encoding

Kun Wang,[§] Ben Cao,[§] Tao Ma, Yunzhu Zhao, Yanfen Zheng, Bin Wang,* Shihua Zhou,* and Qiang Zhang



Cite This: *J. Chem. Inf. Model.* 2024, 64, 1719–1729



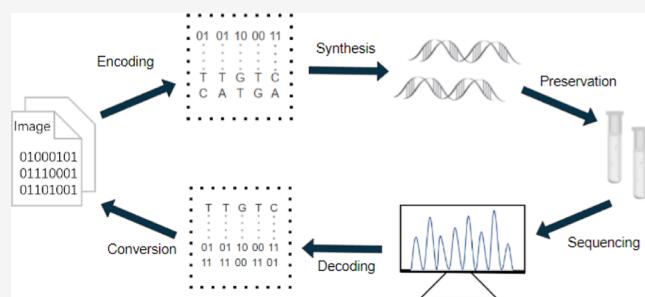
Read Online

ACCESS |

Metrics & More

Article Recommendations

ABSTRACT: Current DNA storage schemes lack flexibility and consistency in processing highly redundant and correlated image data, resulting in low sequence stability and image reconstruction rates. Therefore, according to the characteristics of image storage, this paper proposes storing images in DNA via base128 encoding (DNA-base128). In the data writing stage, data segmentation and probability statistics are carried out, and then, the data block frequency and constraint encoding set are associated with achieving encoding. When the image needs to be recovered, DNA-base128 completes internal error correction by threshold setting and drift comparison. Compared with representative work, the DNA-base128 encoding results show that the undesired motifs were reduced by 71.2–90.7% and that the local guanine-cytosine content variance was reduced by 3 times, indicating that DNA-base128 can store images more stably. In addition, the structural similarity index (SSIM) and multiscale structural similarity (MS-SSIM) of image reconstruction using DNA-base128 were improved by 19–102 and 6.6–20.3%, respectively. In summary, DNA-base128 provides image encoding with internal error correction and provides a potential solution for DNA image storage. The data and code are available at the GitHub repository: https://github.com/123456wk/DNA_base128.



INTRODUCTION

The International Data Corporation reported that data will reach 175 PB in 2025 and 1.9×10^9 PB by 2040,¹ making large-scale data storage a challenge. DNA, the carrier of genetic information for most organisms, offers advantages over traditional storage mediums, such as a high storage density and an extended storage life span. Therefore, DNA has become one of the most promising storage mediums. DNA storage primarily involves five major processes: encoding, synthesis, preservation, sequencing, and decoding,^{2–4} as shown in Figure 1. Encoding is an important step in DNA storage because reasonable and effective encoding can improve the encoding density and reduce the probability of sequencing errors. In 2012, Church et al.⁵ employed next-generation synthetic sequencing techniques to successfully store 650 KB of data in DNA, verifying that DNA can be stored at a scale and that DNA storage density can be estimated. The following year, Goldman et al.⁶ introduced the rotary encoding method to address the homopolymer constraint issue, which utilizes a ternary conversion and the previous base to determine the current base. The team achieved an encoding rate of 0.33 bits/nt. Although the guanine-cytosine (GC) content constraint could not be satisfied, the rotation concept provided a valuable insight for subsequent development. Li et al.⁷ proposed a lossy encoding scheme for image data by improving rotary encoding, which met GC content and homopolymer constraints and improved

storage density. Furthermore, Blawat et al.⁸ proposed a block encoding method to divide the data into groups of 8 bits. By setting the combination of fixed and optional positions, the method guarantees that the first three bases and the last two bases differ. Compared with rotational encoding, the coupling between data blocks was reduced and the net information density (NID) reached 0.92 bits/nt, but the method could not meet the GC content constraint. Furthermore, Zhang et al.⁹ built upon the block-wise encoding approach by introducing balanced Zhang encoding. This modification satisfied both the GC content and homopolymer constraints. Additionally, this encoding method offers a certain level of error detection capability. In addition to block encoding to reduce the generation of DNA storage errors, Erlich and Zielinski¹⁰ applied fountain codes to DNA encoding. Through an exclusion or operation, the encoding sequence between seeds met the GC content and homopolymer constraints, and the NID was 1.57 bits/nt. To reduce data redundancy, Wang et al.¹¹ further improved the fountain code by replacing the index with partial

Received: October 1, 2023

Revised: December 9, 2023

Accepted: January 18, 2024

Published: February 22, 2024



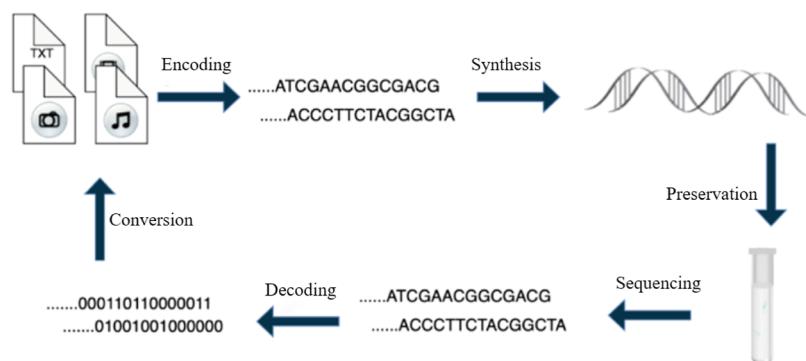


Figure 1. Overview schematic of DNA storage.

data to improve the accuracy of sequencing and decoding. Furthermore, Cao et al.¹² designed encodings using graph convolutional networks and self-attention, achieving high storage density and read-write efficiency.

In DNA storage, because synthesis and sequencing make it difficult to achieve a single-molecule-level operation, one cannot avoid the uncertainty generated in the storage process by optimizing only the encoding scheme. Therefore, to improve the success rate of data decoding and reading accuracy, researchers must introduce error correction schemes.^{13,14} Currently, two main types of error correction approaches exist. The first type involves performing logical operations on the original data and adding error correction codes such as Reed-Solomon (RS) and low-density parity-check (LDPC) at the end.^{15–19} In 2015, Grass et al.²⁰ innovatively applied RS to DNA sequences, utilizing the logical redundancy of the data to achieve error detection and correction within sequences. Shufang and Kang²¹ employed LDPC codes for data encoding, combining the characteristics of the encoded data with RS error correction codes to achieve data correction capabilities. However, a limitation of the study was the occurrence of sequence loss, implying that some data could be lost during the correction process. To further enhance the error correction performance, Meiser et al.²² proposed a dual-code error correction scheme. Both the inner and outer codes utilize RS codes. The inner code is primarily employed to correct data payload errors, while the outer code is utilized to correct index sequence errors, thus preventing the loss of sequences. Another common approach is to enhance data error correction capabilities by introducing constrained codes, such as Hamming distance, HMM, BCH, Levenshtein, and barrier.^{23–30} Li et al.²⁶ introduced the Levenshtein distance in their research and combined it with the GC content balance and specific position correction codes to propose the DNA-LC encoding scheme. This error correction encoding scheme can achieve 100% correction for single errors and correct for two errors under specific conditions. However, this scheme cannot correct for multiple errors. To further improve the correction ability of single-strand multiple errors, Zan et al.²³ proposed a layered error correction strategy (ECS) for DNA storage. By introducing the minimum hamming distance, the researchers constructed a 6-bit DNA sequence table. When one or two errors exist in the 6-bit DNA sequence, the error correction is realized according to the difference between the sequence blocks. In 2021, Dimopoulou et al.³¹ introduced DWT transform to realize image compression by quantizing wavelet sub-bands. The class of code word converted into a base sequence is misjudged according to the known functions in the decoder, and if an error occurs, it is converted to

the code word closest to the correct class. Finally, the Lena image is stored in vitro. The team thus achieved image reconstruction with a peak signal-to-noise ratio (PSNR) of 14.3. To further enhance image reconstruction success, in 2022, Li et al.³² proposed a barrier strategy. This strategy introduces "AA" on fixed-length bases to reduce the high correlation between rotated encoding data, thereby minimizing the data displacement caused by insertion and deletion errors. With an error rate of 0.5%, the structural similarity index (SSIM) of the reconstructed image reached 0.5528. Furthermore, Pan et al.³³ introduced the Hilbert space-filling curve, differential encoding, and Huffman coding to achieve image compression, and they integrated LDPC error correction encodes at the local feature level of the image. During decoding, automatic color change detection and unequal error protection were employed for image reconstruction. With an error rate of 0.5%, the SSIM of the reconstructed image reached 0.67. In data security, Hemalatha et al.³⁴ greatly improved the performance of an image steganalyzer by using the third-order SPAM feature and integrated classifier. On the premise of ensuring image quality, Sahu et al.¹⁹ improved the security of images by improving LSB steganography.

When processing image data with high redundancy and high correlation, current DNA storage schemes have shortcomings in flexibility and consistency, resulting in decreased sequence stability and a low image reconstruction rate. Therefore, this paper proposes storing images in DNA via base128 encoding to improve sequence stability and image reconstruction quality. In the data writing stage, the data are first segmented and probability statistics are generated according to the characteristics of the data. Then, the greedy algorithm is used to optimize the encoding set that meets various constraints and dynamically map the data block to realize DNA-base128 encoding. In terms of data reading, the data can be judged by setting the threshold and performing node and block drift comparisons. Then, according to the combination of constraints for filtering and probability selection for internal error correction, DNA-base128 can realize a high-quality image reconstruction. The encoding sequence not only satisfies the conventional constraints but also achieves outstanding performance in local GC content and undesired motifs. Compared with the representative work, the undesired motifs were reduced by 32.4–41.8%, and the local GC variance was reduced by 3 times. In the case of adding the same errors, the reconstructed image had less noise and a higher structural similarity. Specifically, after the addition of a 1% error, the quantitative index SSIM and MS-SSIM of the DNA-base128 scheme-reconstructed images were increased by 43–97 and 7.2–30.2%, respectively. DNA-base128 offers internal error

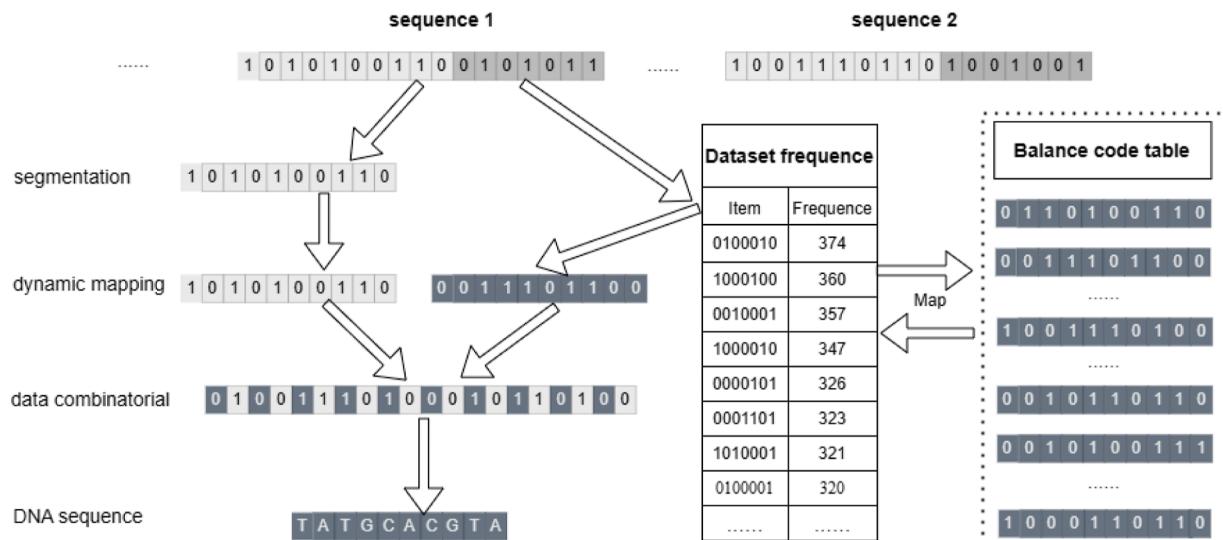


Figure 2. Pipeline of the DNA-base128 encoding process.

correction of image storage methods, providing a potential solution for subsequent image storage schemes.

METHODS

Low sequence stability and a poor image reconstruction effect are common problems in the DNA storage of image data. The main reason is that current schemes lack flexibility and consistency for image data with high redundancy and high correlation.²⁹ To solve these problems, this work proposes storing images into DNA via base128 encoding. In the data writing phase, the original data are first divided and probability statistics are generated. Then, the coding set sequence sets weights for the GC content, homopolymer, and occurrence probability of undesired motifs through a greedy algorithm. The weights are sorted from small to large, and dynamic mapping is realized by combining the probability statistics of data blocks. In the data read phase, first by data drift, the threshold comparison, combination constraint, and probability selection are used to realize the internal error correction. Then, the data decoding is completed according to the base mapping rules to achieve a high-quality image reconstruction.

Data Segmentation and Probability Statistics. DNA encoding transposes each pixel of an image into binary data and stitches it into a binary data set. Depending on the probability of the data blocks, different mapping rules are provided for flexible and efficient encoding. First, the data block is segmented by setting a fixed length, and the frequency statistics are generated according to the segmented data block. Here, the data are divided into a set of 17-bit data sequences. Second, each DNA sequence is further subdivided into odd and even modules, where the first 10 bits of binary data are assigned to even data blocks and the last 7 bits of data are assigned to odd data blocks. According to the frequency of occurrence in the data set, the odd data block has a one-to-one correspondence with the constructed balanced code table, and the mapping relationship is dynamically updated to improve the base utilization of the encoding block. Finally, the odd data blocks are combined by the balance code of the mapping table, and the even data blocks are combined one by one to complete the data transformation.

DNA-base128 Encoding. The accuracy of DNA sequencing is not only related to GC content and homopolymer

constraints but also affected by the combination of bases.^{35–37} Some specific base combinations such as “GAC,” “CAC,” “GTC,” “GTG,” “GCG,” “CGC,” “TCT,” “ACT,” “AGA,” “ATA,” “TAT,” and “TGC” have a higher probability of error than other combinations.^{38,39} Therefore, the frequency of undesired motifs should be reduced in the encoding sequence to decrease the error rate during sequencing. When constructing the DNA-base128 encoding table, first, all combinations of 10-bit binary data blocks are obtained, and the data blocks that meet the “01” balance and do not contain four or more consecutive identical binary data are preliminary screened. Second, the greedy algorithm is used to calculate and label the probability of undesired motifs and three base occurrences for the preliminarily screened sequences. For example, in the encoding process of “101,” four undesired motifs, “GAC”, “CAC”, “GTC”, and “GTG” appear, and the probability is marked as 4. Finally, the labeled data blocks are sorted from small to large, and the first 128 bits of the data blocks are filtered to complete the construction of the DNA-base128 encoding table.

The complete encoding flow of DNA-base128 is shown in Figure 2. First, according to the statistical frequency, the DNA-base128 encoding table is associated with the data blocks sorted from small to large, and an array of probability-data-balanced codes is formed. For different data sets, the mapping relationship is dynamically updated according to the change of data block probability with the encoding table. We reduce the frequency of undesired motif combinations in sequences, thus reducing the probability of errors in DNA sequencing. Then, the odd part is transformed into the corresponding balanced code and combined with the even part one by one. Finally, according to the base mapping rules (00-A, 01-T, 10-C, 11-G), the binary is converted into the DNA sequence.⁴⁰

Error Correction and Decoding. The above encoding schemes reduce the error rate of synthesis and sequencing by adding a variety of biological constraints, but they cannot completely avoid error generation. Therefore, we must carry on the error checking and correction in the decoding process.²⁶ Current error correction methods realize error correction through physical redundancy or logical redundancy (RS, LDPC),⁴¹⁻⁴³ which both cause data redundancy. DNA-base128 works through the data characteristics of balance codes and the limitations of DNA-base128 encoding tables, so

that the encoding scheme has strong robustness in the face of errors. When an error occurs, the error judgment is realized by setting a threshold and comparing the drift of nodes and blocks, and the data are corrected according to combination constraint screening and probability comparison. The following is a detailed description of the error detection, substitution, insertion, and deletion methods.

Error Detection. The inspection of the error position is the first step in error correction. The common types of errors in DNA storage include substitution, insertion, and deletion. Insertion and deletion cause a shift in the data and have a great impact on the overall data. Therefore, data drift is introduced into DNA-base128 to improve the ability of error type recognition. Drift is mainly divided into node drift and block drift. Node drift refers to the shift of the current data block by about one base. Block drift means that in the current interval, data blocks are retrieved and compared in groups of 10.⁴⁴ Block drift helps error detection algorithms identify substitution errors and replacement errors, and node drift can further identify insertion errors and deletion errors in substitution errors. In addition, the set $Dic(T, S, N, \text{and } P)$ is constructed in this scheme. T is a DNA-base128 encoding table that satisfies the GC content balance, homopolymer constraint, and restriction of undesired motifs through the greedy algorithm. S is represented as a 7-bit binary data block table, N is the length of the dictionary, and P is the frequency of occurrence of 7-bit binary data. The flow diagram of the error checking algorithm is shown in Figure 3.

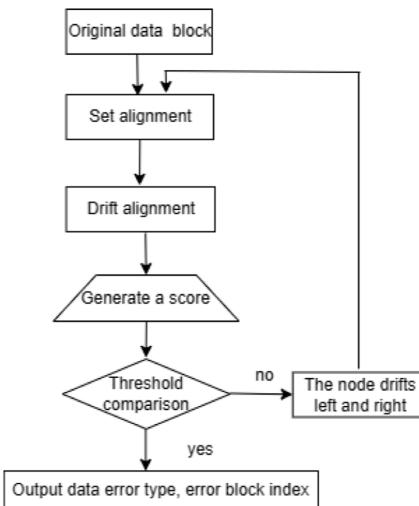


Figure 3. Schematic diagram of the DNA-base128 error determination process. By satisfying the drift ratio on the number of relations with a predetermined threshold, we obtain the error place and type.

Given the positive integer l , the odd block $s = 0, 1^l$. Data blocks in the consideration of constraints include the “01” balance, consecutive identical binary data less than 4, and a data block frequency greater than 1. Here, p_1 refers to the number of occurrences of the data block s in the data set, and the data block that satisfies the constraint can be expressed as

$$Dic(s) = (0, p_1), \quad p_1 > 0 \quad (1)$$

When the data do not meet the above constraints, we need to further confirm the type of error. q denotes the index position of the data block s , U represents the length of the DNA sequence,

and the number of times t the data blocks need to be traversed is expressed as

$$t = \begin{cases} 6, & U/10 - q > 6 \\ U/10 - q - 1, & U/10 - q \leq 6 \end{cases} \quad (2)$$

After the data blocks are traversed downward and compared, the number of data blocks that meet the constraint is n , which can be expressed as

$$n = Block_D(q, t, Odd_MaxGroup) \quad (3)$$

If $n \geq t - 1$, it is a replacement error, and Algorithm 2 is used to rectify the error. Otherwise, it is judged as a translation error (insert and delete), and the new DNA sequence $Odd_MaxGroup_z$ is obtained by drifting nodes left and right, where $z = 0$ indicates left drift, and $z = 1$ indicates right drift. Then, the data block traversal alignment is performed t times.

$$Odd_MaxGroup_z = Note_D(q, t, z, Odd_MaxGroup) \quad (4)$$

Here, if $t = n$, it is judged as an insertion error. Otherwise, to delete the error, we implement translation error correction in Algorithm 3. The specific process is shown in Figure 4, and the detailed process of error judgment is shown in Algorithm 1.

Algorithm 1: Error_judgment

```

Input: s, q, Odd - MaxGroup, dictionary
Output: sus/ins/del, q, s
1 if q + 6 <= Len(Odd - MaxGroup) then
2   | x = 6, y = 5
3 else
4   | x = Len(Odd - MaxGroup) - q - 1, y = x - 1
5 end
6 if y < Block_D(q, x, Odd - Maxgroup) and Odd - MaxGroup[q] ≠ dictionary
then
7   | Output sus, q, s
8 else
9   | Odd - Maxgroup1 = Note_D(q, z, Odd - Maxgroup);
10  | if Block_D(q, x, Odd - Maxgroup1) = x then
11    |   | Output ins, q, s
12  | else
13    |   | Output del, q, s
14 end
15 end
  
```

To better explain the error-judgment algorithm, the section uses a simple demo to illustrate it. Suppose that a sequence insertion error occurs and the DNA sequence becomes “GCATGCGAAT CTCCAGATGC AGATCAGGTC ATTCGCAACA GAGTTGGCAC AA GTCTTGAC CGCTCGAACGT.” First, in the process of data error correction, the DNA sequence is converted to binary data and divided into odd and even data. Second, the combination constraints are used to determine whether the current data block has errors. As shown in the data above, the first data block does not conform to the “01” balance constraint; that is, there is an error in the data block and the position of the base block where the error is located can be inferred. Then, the type of error is preliminarily judged by the downward block drift traversal, and the threshold is set. If the data block drifts downward six times and the data block that meets the combination constraint is lower than the threshold for 1 time, it can be inferred that the data block has an insertion or deletion error. Finally, the error type is further determined by moving to the left and right nodes and then recomparing the block with block drift. By performing left-node shifting and conducting six downward traversals, for data blocks that satisfy the combination constraint, their count is 6, exceeding the preset threshold. Therefore, it can be inferred that the first data block has experienced an insertion error.

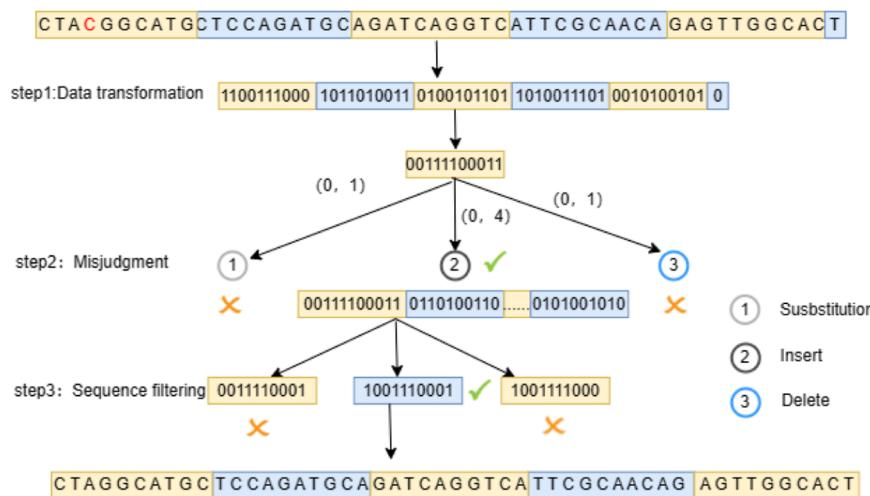


Figure 4. Insert error correction process based on internal error correction. First, DNA sequence segmentation is performed for the module of binary data. Then, in Algorithm 1, the threshold value is set according to the drift ratio and the type and location of the error determined. Finally, the internal error correction is carried out by combination constraint screening and probability comparison.

Algorithm 2: Substitution error correction

```

Input: s, q, Odd - MaxGroup, dictionary
Output: s1
1 if Loc(0) > 5 then
2   switch L(s, 0) = 6 do
3     | s1 = Group_P(s, h + 2, h + 3)
4   end
5   case L(s, 0) = 5 do
6     | s1 = Group_P(s, h + 2, h + 3, h + 4)
7   end
8   case L(s, 0) = 4 do
9     | s1 = Group_P(s, h, h + 1, h + 2, h + 3)
10  end
11 otherwise do
12   | while i < 10 do
13     |   if s(i) = 0 then
14       |     | s1 = Pro_comp(Group_P(s, F(0), Group_P(s, i)))
15     |   end
16   | end
17 end
18 else
19   | Judge the final data in the same way
20 end

```

Substitution Error. When Algorithm 1 detects an error, it receives information containing the data block, the location of the data block, and the type of the error. Assume that s is an error. $Group_P(s, i, j)$ represents the i -th and j -th replacement of s , and the data block with the largest correct rate is obtained according to the GC content balance, homopolymer constraint, and data probability screening. $L(s, b)$ is the length of a block with two or more consecutive identical b values, where b denotes 0 or 1. h indicates the index position of the same base in s . $Loc(b)$ is denoted as the positions where two or more consecutive “ b ” occur, $Pro_comp(b)$ indicates sequence alignment, $F(b)$ represents the index where b appears in s , and the specific process is shown in Algorithm 2.

To better explain the substitution error correction algorithm, the section uses a simple demo to illustrate it. Assume that data change from “AGCTATCGAC” to “CGCTATCGAC.” First, the replacement error of the data block is determined by Algorithm 1. Second, the data blocks are converted to odd data “1110001101” and even data “0101010100.” According to the balance constraint of odd-numbered data “01” and the odd-numbered data block being composed of six 1s and four 0s, it is inferred that the data block has been replaced by 0 to 1. According to the combination constraint satisfied by the odd data, four consecutive equal binary combinations do not occur, position 1 in the third and seventh positions does not have a conversion error, and the sequence satisfying the condition is

filtered. On this basis, the sequence is further filtered by whether it exists in the encoding table and whether the probability of occurrence is greater than 0. Finally, the remaining sequences are compared according to their frequency of occurrence in the data set, and the sequence with the largest probability is selected as the correct data. Moreover, according to the error index dual data block probability alignment, the final dual data block is obtained.

Insertion and Deletion of Errors. When Algorithm 1 detects an insertion or deletion error, it receives information about the DNA sequence, data block, index location, and error type. $Delete_p(s, h_1, h_2)$ means that s deletes data at h_1 -th and h_2 -th, respectively. According to GC content balance, homopolymer constraint, and data probability screening, the most accurate data block is obtained. $Insert(s, h_1, h_2)$ means that s inserts data in h_1 -th and h_2 -th, respectively, finding the data block with the highest accuracy through multiple constraints and probability comparisons. Finally, according to the corrected data and index, we change the location of the dual data for the same operation; the specific process, as shown in Algorithm 3.

Algorithm 3: Insert and delete errors

```

Input: s, q, Odd - MaxGroup, dictionary, ins, del
Output: s1
1 if ins then
2   switch Loc(0) > 5 and L(s, 0) = 4 do
3     | s1 = Delete_P(s, h)
4   end
5   case Loc(0) > 5 and L(s, 0) = 3 do
6     | Delete_P(s, h, h + 1, h + 2, h + 3)
7   end
8   otherwise do
9     | while i < len(s) do
10    |   if s(i) = 0 then
11      |     | s1 = Pro_comp(Delete_P(s, F(0), Delete_P(s, i)))
12    |   end
13   | end
14 end
15 else
16   switch Loc(0) > 5 and L(s, 0) = 5 do
17     | s1 = Insert(s, h + 2, h + 3)
18   end
19   case Loc(0) > 5 and L(s, 0) = 4 do
20     | s1 = Insert(s, h + 2, h + 3, h + 4)
21   end
22   otherwise do
23     | s1 = Pro_comp(Insert_P(s, F(0), Insert_P(s, i)))
24   end
25 end

```

To better explain the insert-and-delete error correction algorithm, the section uses a simple demo to illustrate it.

Suppose that the data change from the “AGTGCGATTA” data block to “AGTGCCGATTA.” First, according to the error type of Algorithm 1, the insertion error of the data block is determined, and the index position of the error data block is obtained. Second, the error data block is converted into odd data “0101110100” and even data “01110010010.” Filter was according to the constraints of constructing the encoding table. As shown in the data presented above, the data block that meets the combination constraints cannot have four consecutive identical binary data. Therefore, the odd data block is “0101110101,” and the insertion error may occur in bits 4, 5, 6, and 7 of the even data block. The final result is obtained according to the probability comparison to complete the error correction.

DNA-base128 Decoding. When data are read, the DNA sequence is first divided according to a certain number of bases, and then, the segmented DNA sequences are decoded by base mapping rules (A-00, T-01, C-10, G-11) to obtain even module data and odd module data. Then, a 7-bit Odd-backward is obtained through the probability-data-balanced set. For example, the odd-numbered data block is “0111000110,” and the Odd-backward is “0101011” through the probability-data-balanced table. Finally, the even data block comes before Odd-backward, and the data are recombined to complete the data decoding.

RESULTS

To verify the performance of DNA-base128, we compared and analyzed four aspects: encoding results, local DNA sequence performance, undesired motifs, and image reconstruction. First, the methods in this paper were compared for GC content, ECS, error correction redundancy (CEC), and NID. Then, local GC content was compared because more stable local GC content could reduce sequencing coverage.³³ Second, the combination of undesired motifs can explain the sequencing error rate. Finally, in terms of image reconstruction, our method was compared with the four encoding schemes of Li,⁷ Zhang,⁹ Church,⁵ and Blawat⁸ in the six indexes of PSNR, SSIM, FSIM, MS-SSIM, UQI, and MSE. The results show the noise of the reconstructed image and the performance of structural similarity as well as verify the reconstruction effect of the image under different error rates.

General Performance Analysis. To verify the DNA-base128 encoding performance, we compared the GC content, ECS, CEC, and NID with the current representative work,^{5,7–9} as shown in Table 1. ECS and CEC had a profound impact on

Table 1. DNA Storage General Performance Comparison

method	GC	ECS	CEC	ND
Church ⁵	yes	repetition	redundancy	0.83
Blawat ⁸	no	forward error correction	redundancy	1.03
Zhang ⁹	yes	RS	redundancy	1.58
Li ⁷	yes	barrier correction	redundancy	1.85
this work	yes	inner-error correction	no	1.61

the difficulty of error data recovery and encoding density. Compared with the other external error correction methods, such as forward error correction, RS, and barrier correction, the internal error correction scheme of DNA-base128 could avoid the reading failure caused by the error correction code and reduce the redundancy of error correction, thus improving the data reading efficiency and encoding density. The NID is the

ratio of binaries to used bases. The net encoding density of DNA-base128 has significant advantages compared with Blawat,⁸ Zhang,⁹ and other schemes. Compared with the Li⁶ scheme, DNA-base128 had a slight shortage in NID but made up for the defects of the local GC content imbalance, undesired motif limitation, and loss decoding.

Local Performance Analysis of DNA Codes. During the reading of data stored in DNA, the base composition of the DNA sequence had an important impact on the results of DNA sequencing.⁴¹ Compared with the deviated GC content sequence, the balanced GC content encoding sequence had higher coverage and less of a secondary structure, which is promising to achieve higher-quality reads under the same sequencing coverage.^{35,45,46} Moreover, DNA sequences locally satisfying the GC content balance can further improve the accuracy of data reading.¹¹ To validate whether the sequences adhere to the local GC content balance requirement, we randomly selected equidistant base sequences from the Lena image data. These sequences were further divided into a sequence of 15 bases in length to make a comparison of GC content with the schemes of Li⁷ and Blawat.⁸ As shown in Figure 5a, the blue, yellow, and red lines represent the local GC content of the Li⁷, Blawat,⁸ and DNA-base128 schemes, respectively. The maximum and minimum values of the lines show that DNA-base128 fluctuated less. In addition, the local variance of the DNA-base128 encoding sequence was 0.0042, and the GC content of the sequence was between 45 and 55%. The variances of the local GC content of Li⁷ and Blawat⁸ were 0.0202 and 0.0126, respectively. The experimental results show that the DNA-base128 encoding sequence had a more balanced GC content than the above two encoding sequences. That is, under the same sequencing coverage, higher-quality and higher-accuracy data readings could be achieved.

Comparison of Undesired Motifs. The accuracy of DNA sequencing is not only related to the GC content and homopolymer but is also affected by the base combination in the sequence.⁴⁶ Compared with A and C, G and T have a high probability of random errors. Even different combinations of the same base present different error rates, such as “GAC,” “CAC,” “GTC,” “GTG,” “GCG,” “CGC,” “TCT,” “ACT,” “AGA,” “ATA,” “TAT,” and “TGC,” have a higher error probability than the other base combinations.³⁹ Therefore, the error rate of the data reading can be further reduced by limiting the frequency of undesired motifs in the encoding process. To illustrate the potential of the DNA-base128 method in reducing sequencing error rates, we made a comparison of the occurrence frequency of undesired motifs using the Lena image data set, against the Li⁷, Blawat⁸ schemes. As shown in Figure 5b, DNA-base128 was compared with two encoding schemes at 5000, 10,000, and 20,000 nt. The undesired motifs were reduced by 71.2–90.7% compared with the two encoding schemes. Specifically, when the sequence length was 5000 nt, the undesired motifs were 90% lower than those of Blawat’s scheme. The experimental results show that DNA-Base128 has a good limitation on the undesired motif, which improves the stability of DNA sequence.

Image Reconstruction Effect. To validate the image reconstruction effectiveness, the DNA-base128 scheme incorporates metrics such as PSNR, SSIM, MSE, FSIM, MS-SSIM, and UQI for comparing the reconstructed images^{19,34,47–49} and compares the results with those of the representative work. PSNR, MSE, and UQI are used to measure the difference between the reconstructed image and the original image pixels. SSIM, MS-SSIM, and FSIM are introduced to comprehensively

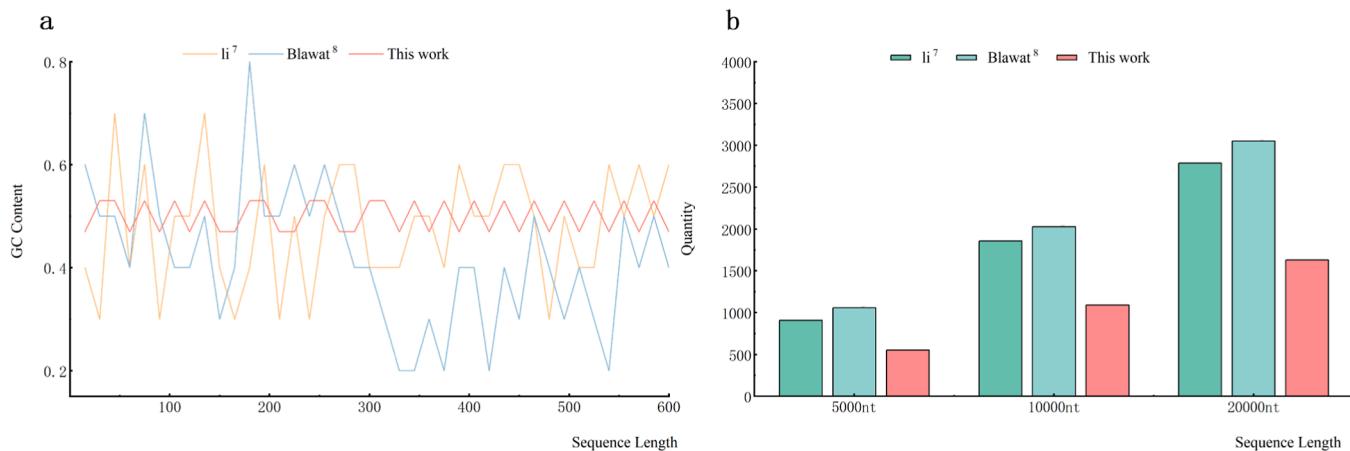


Figure 5. Comparison of the properties of DNA sequences by Li,⁷ Blawat,⁸ and DNA-base128 encoding. (a) Comparison of local GC content. The blue, yellow, and red lines represent the local GC content of the Li,⁷ Blawat,⁸ and DNA-base128, respectively. Balanced local GC content indicates higher sequencing coverage and less secondary structure for DNA sequencing. (b) Comparison of the content of undesired motifs. The reduction of the content of undesired motifs could improve the accuracy of data reading.

evaluate the image reconstruction effect in terms of the structural information and texture features of the image. Note that PSNR, SSIM, FSIM, MS-SSIM, and UQI values are proportional to the reconstruction effect, while the MSE value is inversely proportional to the reconstruction effect. The default error rate of substitution, insertion, and deletion was 8:1:1, and the length of the DNA strand was 150 nt.¹⁰ In order to evaluate the performance of the DNA-base128 scheme on various image reconstructions, the DNA-base128 scheme selected three types of images for experimental comparison. These included a gray image (Lena) and two black and white color images from the USC-SIPI.⁵⁰ The study simulated the image reconstruction quality under different error rates by introducing random DNA sequence errors at rates of 0.25, 0.5, 0.75, and 1%, as shown in Figure 6.

According to gray images of PSNR, SSIM, MSE, FSIM, MS-SSIM, and UQI, DNA-base128 had better performance under the condition of a low error rate. With the increase of the error rate, compared with other representative work, the change trend of PSNR, SSIM, MSE, FSIM, MS-SSIM, and UQI of DNA-base128 was slower, which indicates that DNA-base128 has a better image reconstruction effect in black and white images. Note that when a 1% error is introduced into gray images, the PSNR, SSIM, FSIM, MS-SSIM, and UQI values of DNA-base128 scheme-reconstructed images are 36.18, 0.40, 0.64, 0.71, and 0.89, respectively, and the MSE value is 16. However, the PSNR, SSIM, FSIM, MS-SSIM, and UQI values of other encoding schemes are lower than 33.41, 0.30, 0.61, 0.61, and 0.87, respectively, and the MSE value was higher than 29. To further explore the image reconstruction performance of DNA-base128, black and white images and color images were also compared, as shown in Figure 6. For black and white images, the error correction performance of the five encoding schemes is improved. Compared with other representative works, DNA-base128 achieves the best results in PSNR, SSIM, MSE, and UQI of the reconstructed image and the original image. Although DNA-base128 is lower than Church⁵ and Li's⁷ scheme in the quantitative metric MS-SSIM, the overall image reconstruction effect is better than that of Church⁵ and Li's⁷ encoding scheme. For color images, the correction performance of the five encoding error correction schemes decreases, but compared with other representative works, DNA-base128

achieves the best results in PSNR, SSIM, MSE, FSIM, MS-SSIM, and UQI, and the reconstruction effect is better. For example, when the error of 1% is added to the color image, the PSNR, SSIM, FSIM, MS-SSIM, and UQI values of DNA-base128 scheme-reconstructed images are 32.43, 0.34, 0.64, 0.67, and 0.78, respectively, and the MSE value is 37.13. The PSNR, SSIM, FSIM, MS-SSIM, and UQI values of other encoding schemes are lower than 32.09, 0.24, 0.61, 0.58, and 0.74, and the MSE value is higher than 38.36.

In the process of image reconstruction, DNA-base128 filters the candidate data by using GC equalization, biological constraints, and encoding table constraints and finally realizes the maximum probability correction of data by probability comparison. Therefore, there will be some errors in the process of image reconstruction. In order to further explore the influence of errors in image reconstruction, multiple sets of errors were randomly added to the image. The corrected image is compared with the image without the corrected error by PSNR. For black and white, gray, and color images with error rates of 0–1%, the average PSNR error values of DNA-base128-reconstructed images are 0.861, 0.864 and 0.866, respectively. The experimental results show that the random errors of black-and-white and color images have little influence on the reconstructed image results, and the random errors of gray images have a great influence on the reconstructed image results.

Although DNA-base128 has errors in the process of image reconstruction, the effect of image reconstruction is still better than other representative schemes. There are two main reasons for the significant advantages of the DNA-base128 reconstruction effect. First, according to the data characteristics of high redundancy and high correlation of image data, this error correction scheme can accurately find the position of insertion, deletion, and replacement through a data drift and threshold comparison, reduce the impact of a data shift on the following data, and further reduce the error through probability correction. Second, DNA-base128 weakens the coupling between blocks through block mapping encoding. The encoding rules within a block are simple, which reduces the correlation between data and yields high data robustness. Blawat⁸ and Zhang⁹ utilized block-constraint encoding to process data. However, the encoding rules within the blocks are relatively complex, and a high level of correlation exists among the data.

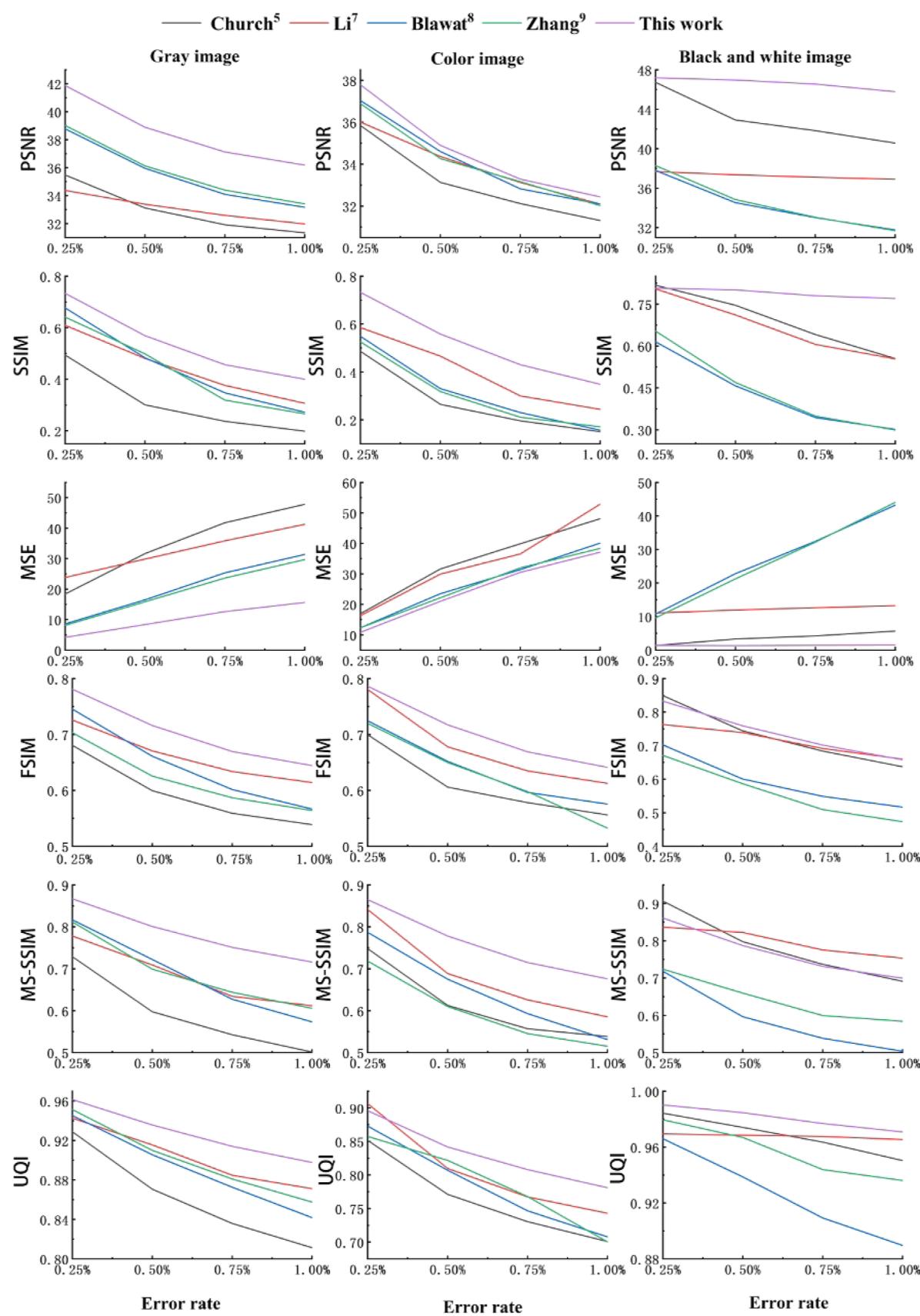


Figure 6. Comparison of DNA-base128 image reconstruction with representative work. Errors of 0.25, 0.5, 0.75, and 1% were added to the transmission channel, and black, red, blue, green, and purple represent the five encoding schemes of Church,⁵ Li,⁷ Blawat,⁸ Zhang,⁹ and DNA-base128, respectively. The reconstructed image was evaluated by PSNR, SSIM, MSE, FSIM, MS-SSIM, and UQI.

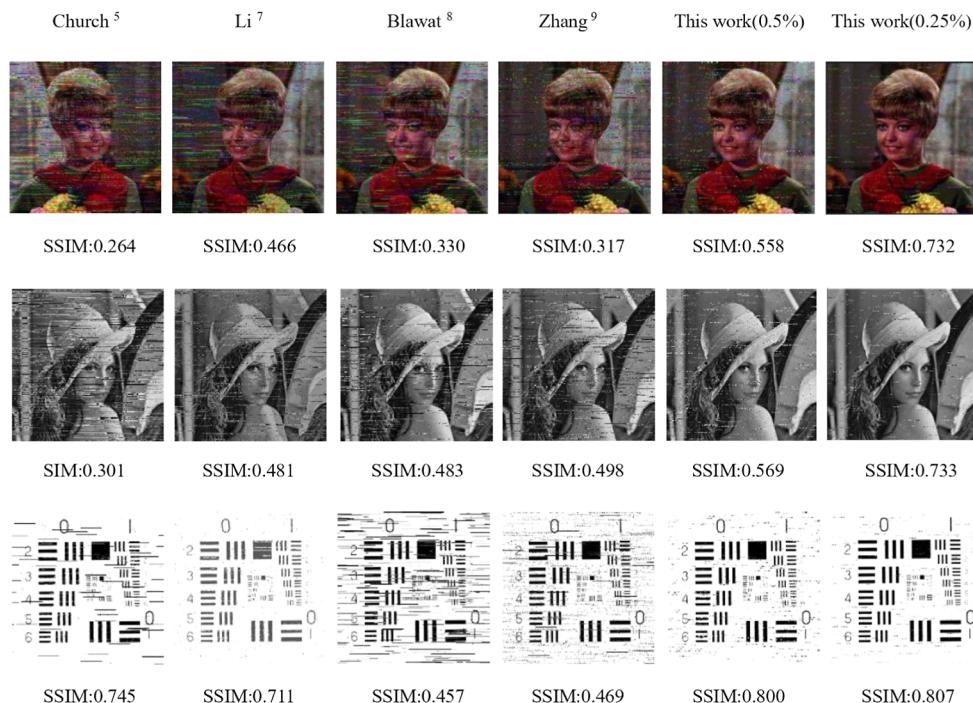


Figure 7. Visual results of DNA-base128 image reconstruction with representative work. The first five columns represent the image reconstruction results of three types of images with randomly introduced errors of 0.5%, while the last column represents the image reconstruction results of three types of images with randomly introduced errors of 0.25%. Among them, the error types of insertion, deletion, and substitution account for 80, 10, and 10% respectively.

The error correction performance showed similar results under different errors, but the image reconstruction performance was not high. Blawat⁸ could not meet the GC content constraint, so the reconstruction effect was general in the actual image. For Li,⁷ the rotational scheme is used to encode the data with high correlation, but the barrier strategy is used to reduce the error propagation problem. Thus, the image reconstruction performance is better than that of Blawat⁸ and Zhang.⁹ To reflect the effect of image reconstruction more intuitively, the reconstructed images of five encoding schemes when the error is 0.5% are displayed, as shown in Figure 7.

CONCLUSIONS

In order to solve the current storage schemes for image data stored in DNA sequence stability and the problem of low rate of image reconstruction, this paper proposes DNA-base128. In the data writing stage, the mapping relationship is dynamically updated by converting images into binary data to ensure that general constraints and undesired motifs limit are met. To illustrate the encoding performance of DNA-base128, this paper compared the local sequence performance and the undesired motif limit, and DNA-base128 achieved competitive results. The base composition of a DNA sequence is an important factor to determine the success of DNA sequencing, and the balanced distribution of bases can reduce the sequence coverage to generate secondary structures. As shown in Figure 5, the local GC variance of DNA-base128 was reduced by 3 times compared to the results of Li⁷ and Blawat,⁸ and the GC content was between 45 and 55%, indicating excellent base balance characteristics. The results indicate that reads of higher quality can be obtained with the same sequencing coverage. In addition, the accuracy of DNA sequencing is not only related to constraints such as GC content but also affected by the

undesired motifs in the sequence. By comparing our scheme with representative work such as Li⁷ and Blawat,⁸ the undesired motifs were reduced by 71.2–90.7%, indicating that our scheme can reduce the error rate of encoding sequence and improve the stability of encoding sequence. When the image data need to be read, DNA sequences are first judged by setting the threshold and performing node and block drift comparisons, and the internal error correction and decoding are carried out according to the combination constraint screening and probability comparison. Compared with the representative work, the PSNR, SSIM, FSIM, and MS-SSIM values of the DNA-base128 scheme-reconstructed image and the original image are improved by 5.5–21, 19–102, 3.7–20, and 6.6–20.3%, respectively. Especially, after adding the error of 1% to the gray image, the PSNR, SSIM, FSIM, and MS-SSIM values of the DNA-base128 scheme-reconstructed image increased by 8.2–22, 43–97, 4.9–20.7, and 16.3–40%, respectively. The images also have a lower image noise and a higher structural similarity. In summary, DNA-base128 provides an image encoding option with internal error correction according to the characteristics of image data in DNA storage, which provides a potential solution for DNA image storage.

At present, the DNA-base128 scheme can only deal with errors in sparse direction pairs and has no suitable solution for relatively dense errors. In terms of encoding, there is coupling between bases and data blocks, and when errors occur, the scope of errors will be expanded. Therefore, the recovery effect of DNA image storage is not ideal when dealing with high complexity and dense errors. In future work, we will focus on developing storage models with low coupling between sequences, satisfying multiple biochemical constraints (homopolymers, GC equilibrium, secondary structure, and undesired motifs), and sequence adaptive error weights. We will also continue to develop encoding error correction algorithms to

satisfy arbitrary global constraints and dense errors, providing solutions for the high-quality DNA storage of images.

■ ASSOCIATED CONTENT

Data Availability Statement

As described above, https://github.com/123456wk/DNA_base128 is open-source and freely available on the GitHub repository. The data set used to test the functionality of the tool and additional installation information is available on the same repository.

■ AUTHOR INFORMATION

Corresponding Authors

Bin Wang — *The Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian 116622, China; orcid.org/0000-0002-8800-000X; Email: wangbinpaper@gmail.com*

Shihua Zhou — *The Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian 116622, China; Email: zhoushihua@dlu.edu.cn*

Authors

Kun Wang — *The Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian 116622, China*

Ben Cao — *School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China*

Tao Ma — *Brain Function Research Section, China Medical University, Shenyang 110001, China*

Yunzhu Zhao — *The Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian 116622, China*

Yanfen Zheng — *School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China*

Qiang Zhang — *The Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education, School of Software Engineering, Dalian University, Dalian 116622, China*

Complete contact information is available at:

<https://pubs.acs.org/10.1021/acs.jcim.3c01592>

Author Contributions

[§]K.W. and B.C. are joint first authors.

Funding

This work is supported by 111 Project (no. D23006), the National Natural Science Foundation of China (no. 62272079), Natural Science Foundation of Liaoning Province (no. 2022-KF-12-14), the Postgraduate Education Reform Project of Liaoning Province (no. LNYJG2022493), the Dalian Outstanding Young Science and Technology Talent Support Program (no. 2022RJ08), and the Fundamental Research Funds for the Central Universities under grant no. DUT23YG122.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank the Key Laboratory of Advanced Design and Intelligent Computing for providing the necessary facilities and equipment to carry out our experiments. The authors also appreciate the guidance and assistance provided by the lab technicians and staff members throughout the experimental process.

■ REFERENCES

- (1) Rydning, D. R.-J. G.-J.; Reinsel, J.; Gantz, J. *The Digitization of the World from Edge to Core*; International Data Corporation: Framingham, MA, 2018; Vol. 16, pp 1–28.
- (2) Zheng, Y.; Cao, B.; Wu, J.; Wang, B.; Zhang, Q. High Net Information Density DNA Data Storage by the MOPE Encoding Algorithm. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2023**, *20*, 2992.
- (3) Organick, L.; Chen, Y.-J.; Dumas Ang, S.; Lopez, R.; Liu, X.; Strauss, K.; Ceze, L. Probing the physical limits of reliable DNA data retrieval. *Nat. Commun.* **2020**, *11*, 616.
- (4) Wang, J.; Zhang, X.; Shi, P.; Cao, B.; Wang, B. A DNA Finite-State Machine Based on the Programmable Allosteric Strategy of DNAAzyme. *Int. J. Mol. Sci.* **2023**, *24*, 3588.
- (5) Church, G. M.; Gao, Y.; Kosuri, S. Next-generation digital information storage in DNA. *Sci.* **2012**, *337*, 1628.
- (6) Goldman, N.; Bertone, P.; Chen, S.; Dessimoz, C.; LeProust, E. M.; Sipos, B.; Birney, E. Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature* **2013**, *494*, 77–80.
- (7) Li, Y.; Du, D. H.; Ou, L.; Li, B. HL-DNA: A Hybrid Lossy/Lossless Encoding Scheme to Enhance DNA Storage Density and Robustness for Images. 2022; *IEEE 40th International Conference on Computer Design (ICCD)*, 2022; pp 434–442.
- (8) Blawat, M.; Gaedke, K.; Hüttner, I.; Chen, X.-M.; Turczyk, B.; Inverso, S.; Pruitt, B. W.; Church, G. M. Forward error correction for DNA data storage. *Procedia Comput. Sci.* **2016**, *80*, 1011–1022.
- (9) Zhang, Y.; Kong, L.; Wang, F.; Li, B.; Ma, C.; Chen, D.; Liu, K.; Fan, C.; Zhang, H. Information stored in nanoscale: Encoding data in a single DNA strand with Base64. *Nano Today* **2020**, *33*, 100871.
- (10) Erlich, Y.; Zielinski, D. DNA Fountain enables a robust and efficient storage architecture. *Sci.* **2017**, *355*, 950–954.
- (11) Wang, P.; Mu, Z.; Sun, L.; Si, S.; Wang, B. Hidden addressing encoding for DNA storage. *Front. Bioeng. Biotechnol.* **2022**, *10*, 916615.
- (12) Cao, B.; Wang, B.; Zhang, Q. GCNSA: DNA storage encoding with a graph convolutional network and self-attention. *Iscience* **2023**, *26*, 106231.
- (13) Xu, C.; Zhao, C.; Ma, B.; Liu, H. Uncertainties in synthetic DNA-based data storage. *Nucleic Acids Res.* **2021**, *49*, 5451–5469.
- (14) Zan, X.; Xie, R.; Yao, X.; Xu, P.; Liu, W. A robust and efficient DNA storage architecture based on modulation encoding and decoding. *J. Chem. Inf. Model.* **2023**, *63*, 3967–3976.
- (15) Ren, Y.; Zhang, Y.; Liu, Y.; Wu, Q.; Su, J.; Wang, F.; Chen, D.; Fan, C.; Liu, K.; Zhang, H. DNA-Based Concatenated Encoding System for High-Reliability and High-Density Data Storage. *Small Methods* **2022**, *6*, 2101335.
- (16) Deng, L.; Wang, Y.; Noor-A-Rahim, M.; Guan, Y. L.; Shi, Z.; Gunawan, E.; Poh, C. L. Optimized code design for constrained DNA data storage with asymmetric errors. *IEEE Access* **2019**, *7*, 84107–84121.
- (17) Ezekannagha, C.; Becker, A.; Heider, D.; Hattab, G. Design considerations for advancing data storage with synthetic DNA for long-term archiving. *Mater. Today Bio* **2022**, *15*, 100306.
- (18) Tong, J.; Han, G.; Sun, Y. An Improved Marker Code Scheme Based on Nucleotide Bases for DNA Data Storage. *Appl. Sci. Basel* **2023**, *13*, 3632.
- (19) Sahu, M.; Padhy, N.; Gantayat, S. S.; Sahu, A. K. Shadow image based reversible data hiding using addition and subtraction logic on the LSB planes. *Sens. Imaging* **2021**, *22*, 7.
- (20) Grass, R. N.; Heckel, R.; Puddu, M.; Paunescu, D.; Stark, W. J. Robust chemical preservation of digital information on DNA in silica

- with error-correcting codes. *Angew. Chem., Int. Ed.* **2015**, *54*, 2552–2555.
- (21) Shufang, Z.; Kang, P. DNA information storage technology based on raptor code. *Laser & Optoelectronics Progress* **2020**, *57*, 151701.
- (22) Meiser, L. C.; Antkowiak, P. L.; Koch, J.; Chen, W. D.; Kohll, A. X.; Stark, W. J.; Heckel, R.; Grass, R. N. Reading and writing digital data in DNA. *Nat. Protoc.* **2020**, *15*, 86–101.
- (23) Zan, X.; Yao, X.; Xu, P.; Chen, Z.; Xie, L.; Li, S.; Liu, W. A hierarchical error correction strategy for text DNA storage. *Interdiscip. Sci.: Comput. Life Sci.* **2022**, *14*, 141–150.
- (24) Rasool, A.; Qu, Q.; Wang, Y.; Jiang, Q. Bio-constrained codes with neural network for density-based DNA data storage. *MATH* **2022**, *10*, 845.
- (25) Yujian, L.; Bo, L. A normalized Levenshtein distance metric. *IEEE TPAMI* **2007**, *29*, 1091–1095.
- (26) Li, X.; Chen, M.; Wu, H. Multiple errors correction for position-limited DNA sequences with GC balance and no homopolymer for DNA-based data storage. *Briefings Bioinf.* **2023**, *24*, bbac484.
- (27) Chen, W.; Wang, L.; Han, M.; Han, C.; Li, B. Sequencing barcode construction and identification methods based on block error-correction codes. *Sci. China: Life Sci.* **2020**, *63*, 1580–1592.
- (28) Chee, Y. M.; Kiah, H. M.; Wei, H. Efficient and explicit balanced primer codes. *IEEE Trans. on Inf. Theory* **2020**, *66*, 5344–5357.
- (29) Wu, J.; Zhang, S.; Zhang, T.; Liu, Y. HD-Code: End-to-End High Density Code for DNA Storage. *IEEE Trans. NanoBiosci.* **2021**, *20*, 455–463.
- (30) Park, S.-J.; Park, H.; Kwak, H.-Y.; No, J.-S. BIC Codes: Bit Insertion-based Constrained Codes with Error Correction for DNA Storage. *IEEE TETC* **2023**, *11*, 764–777.
- (31) Dimopoulou, M.; Antonini, M.; Barbry, P.; Appuswamy, R. Image storage onto synthetic DNA. *Sig. Process Image Commun.* **2021**, *97*, 116331.
- (32) Li, B.; Ou, L.; Du, D. Img-dna: approximate dna storage for images. *Proceedings of the 14th ACM International Conference on Systems and Storage*, 2021; pp 1–9.
- (33) Pan, C.; Tabatabaei, S. K.; Tabatabaei Yazdi, S. M. H.; Hernandez, A. G.; Schroeder, C. M.; Milenkovic, O. Rewritable two-dimensional DNA-based data storage with machine learning reconstruction. *Nat. Commun.* **2022**, *13*, 2984.
- (34) Hemalatha, J.; Sekar, M.; Kumar, C.; Gutub, A.; Sahu, A. K. Towards improving the performance of blind image steganalyzer using third-order SPAM features and ensemble classifier. *J. Info. Secur. Appl.* **2023**, *76*, 103541.
- (35) Schwarz, M.; Welzel, M.; Kabdullayeva, T.; Becker, A.; Freisleben, B.; Heider, D. MESA: automated assessment of synthetic DNA fragments and simulation of DNA synthesis, storage, sequencing and PCR errors. *Bioinformatics* **2020**, *36*, 3322–3326.
- (36) Wang, Y.; Noor-A-Rahim, M.; Gunawan, E.; Guan, Y. L.; Poh, C. L. Modelling, characterization of data-dependent and process-dependent errors in DNA data storage. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2023**, *20*, 2147–2158.
- (37) Zhu, D.; Huang, Z.; Liao, S.; Zhou, C.; Yan, S.; Chen, G. Improved bare bones particle swarm optimization for dna sequence design. *IEEE Trans. NanoBiosci.* **2023**, *22*, 603.
- (38) Zhu, Z.; Zhang, Y.; Ji, Z.; He, S.; Yang, X. High-throughput DNA sequence data compression. *Briefings Bioinf.* **2015**, *16*, 1–15.
- (39) Organick, L.; Ang, S. D.; Chen, Y.-J.; Lopez, R.; Yekhanin, S.; Makarychev, K.; Racz, M. Z.; Kamath, G.; Gopalan, P.; Nguyen, B.; et al. Random access in large-scale DNA data storage. *Nat. Biotechnol.* **2018**, *36*, 242–248.
- (40) Rasool, A.; Hong, J.; Jiang, Q.; Chen, H.; Qu, Q. BO-DNA: Biologically optimized encoding model for a highly-reliable DNA data storage. *Comput. Biol. Med.* **2023**, *165*, 107404.
- (41) Yazdi, S. M. H. T.; Gabrys, R.; Milenkovic, O. Portable and error-free DNA-based data storage. *Sci. Rep.* **2017**, *7*, 5011.
- (42) Jeong, J.; Park, S.-J.; Kim, J.-W.; No, J.-S.; Jeon, H. H.; Lee, J. W.; No, A.; Kim, S.; Park, H. Cooperative sequence clustering and decoding for DNA storage system with fountain codes. *Bioinformatics* **2021**, *37*, 3136–3143.
- (43) Sun, Y.; Han, G.; Liu, C.; Wang, Y.; Fang, Y. An asymmetric-error-aware LDPC decoding algorithm for DNA storage. *Sci. Rep.* **2023**, *27*, 32–36.
- (44) Qu, G.; Yan, Z.; Wu, H. Clover: tree structure-based efficient DNA clustering for DNA-based data storage. *Briefings Bioinf.* **2022**, *23*, bbac336.
- (45) Laehnemann, D.; Borkhardt, A.; McHardy, A. C. Denoising DNA deep sequencing data—high-throughput sequencing errors and their correction. *Briefings Bioinf.* **2016**, *17*, 154–179.
- (46) Löchel, H. F.; Welzel, M.; Hattab, G.; Hauschild, A.-C.; Heider, D. Fractal construction of constrained code words for DNA storage systems. *Nucleic Acids Res.* **2022**, *50*, No. e30.
- (47) Wang, J.; Chen, P.; Zheng, N.; Chen, B.; Principe, J. C.; Wang, F.-Y. Associations between MSE and SSIM as cost functions in linear decomposition with application to bit allocation for sparse coding. *Neurocomputing* **2021**, *422*, 139–149.
- (48) Setiadi, D. R. I. M. PSNR vs SSIM: imperceptibility quality assessment for image steganography. *Multimed. Tool Appl.* **2021**, *80*, 8423–8444.
- (49) Kamil Khudhair, S.; Sahu, M.; Kr, R.; Sahu, A. K. Secure Reversible Data Hiding Using Block-Wise Histogram Shifting. *Electron* **2023**, *12*, 1222.
- (50) The USC-SIPI Image Database. <https://sipi.usc.edu/database/database.php>, 1977.

CAS BIOFINDER DISCOVERY PLATFORM™

PRECISION DATA FOR FASTER DRUG DISCOVERY

CAS BioFinder helps you identify targets, biomarkers, and pathways

Unlock insights

CAS A division of the American Chemical Society