A super robust and efficient DNA storage architecture based on modulation encoding and decoding

Xiangzhen Zan[1], Ranze Xie[1], Xiangyu Yao[1], Peng Xu[1*], Wenbin Liu[1*]

1. Institution of Computational Science and Technology, Guangzhou University, Guangzhou 510006, China

*Corresponding authors: gdxupeng@gzhu.edu.cn; wbliu6910@gzhu.edu.cn.

Abstract: Thanks to its high density and long durability, synthetic DNA has been widely considered as a promising solution to the data explosion problem. However, due to the large amount of random base insertion-deletion-substitution (IDSs) errors from sequencing, reliable data recovery remains a critical challenge, which hinders its large-scale application. Here, we propose a modulation-based DNA storage architecture. Experiments on simulation and real datasets demonstrate that it has two distinct advantages. First, modulation encoding provides a simple way to ensure the encoded DNA sequences comply with biological sequence constraints (i.e., GC balanced and no homopolymers); Second, modulation decoding is highly efficient and extremely robust for the detection of insertions and deletions, which can correct up to ~40% errors. These two advantages pave the way for future high-throughput and low-cost techniques, and will kickstart the actualization of a viable, large-scale system for DNA data storage.

Introduction

As the amount of available data grows exponentially, recent developments in synthesis and sequencing technology are making synthetic DNA more attractive as a new data storage medium[1, 2, 3]. Compared with traditional electric/optical/magnetic storage media, DNA synthesis and sequencing are highly error prone processes characterized by aggregated insertions, deletions and substitutions (IDSs)[4]. It is estimated that third-generation synthesis and sequencing technologies may introduce 10–15% errors in the writing and reading processes[5]. Lee Organick[6] reported that over 88% of reads are of incorrect length because of the insertions and deletions caused by synthesis and sequencing. Antkowiak et.al [7] found that of the 15 million sequences they tested, over 99.9% of the original sequences were retrieved with error. Additionally, DNA data corruption can also occur during synthesis, amplification, and sequencing.

In order to minimize error occurrences, most of the prior works focused on complying with certain constraints on the encoded DNA sequences, such as no homopolymers, guanine–cytosine (GC) content of 40% ~ 55%, and no secondary structures. For example, Goldman et al. [8]adopted a 3-base rotation encoding scheme. Randomization by an XOR operation with a pseudorandom binary sequence is a popular strategy utilized by many works[6, 9]. However, there is not yet a universally accepted standard to build upon for large scale storage applications.

To recover data from the distorted sequenced reads, prior works mainly adopted two approaches: physical and logical redundancy. Physical redundancy can be either storing multiple copies of the same sequence, or repeating each substring several times within the same sequence. Either way, data recovery is then accomplished by multiple sequence alignment (MSA)[10, 11] . This strategy is very simple both in encoding and decoding, but comes at the cost of logical density. For instance, the fourfold redundancy[8] only has a logical density of 0.29bit/base, when the theoretical limit is 2bit/base. Another form of physical redundancy is sequencing coverage. However, this too is not sufficient for lossless data recovery in large-scale storage. In the pioneer work of Church et.al[12], they failed to completely restore the encoded information even with a sequencing coverage of 3000×.

Logical redundancy has been widely used in modern communication. The general idea of logical redundancy is to attach to the end of the original data some extra verification information, which is based on a specific generator matrix. When an error happens, discrepancies arise between the original and verification information. Therefore, the verification information, or error correction (EC) codes, can detect or even correct errors. Reed-Solomon Code[9, 13, 14], BCH code[15, 16], Levenshtein Code[17], Hamming code[18], and LDPC code[19, 20] have been applied in DNA storage. Erlich et al. [21] and Anavy et al. [7] used the classic Luby Transform Code to deal with the loss of DNA strands. But since these EC codes were not developed for indels, their effectiveness is limited for DNA sequences and only works when the error rate is low, usually below 5% [13, 15, 22].

Recently, heuristic algorithms shed a new light for solving IDSs. One way is to correct errors using the Inner-Outer code approach[23]. Press et. al. proposed HEDGES (hash encoded, decoded by greedy exhaustive search)where a modified A* algorithm is applied as the inner decoder to correct indels based on the associations between consecutive bits. Lenz et. al. constructed a possible consensus sequence from the multiple sequences inferred by the hidden Markov model (HMM) [24]. Antkowiak et. al [7] used MSA to infer a consensus sequence from each read cluster before the inner and outer decoder. In fact, the MSA provides a maximal likelihood estimation of the original strand based on multiple sequences, which may correct a large part of the errors in the process, especially the frustrating indels. Another approach is to construct a de Bruijn graph using multiple sequences and infer the correct sequence by a graph searching algorithm. Compared with the EC based methods, heuristic algorithms may tolerate higher error rates ranging from 5-15%.

Future large scale deployment of DNA storage may face an even more error prone

environment, as synthetic biotechnologies will most likely have higher throughput at lower cost [7]. Meanwhile, stored data may suffer from some unpredicted or malicious damages, like the hard disk failures in computers. In addition, DNA molecules may undergo degradation and break over time. To cope with these considerations, it is necessary to develop robust techniques which can meet the challenge of more complicated settings (e.g. 20~30% error rates).

In the field of telecommunication, modulation has accomplished reliable signal transmission by superimposing baseband signals (or low frequency signals) to the carrier signals (or high-frequency signals) [25, 26, 27]. The modulated signal not only has the power to anti-interference, but also has no effect on the modulating signal. In fact, the carrier signals serve as the guideline to protect the modulating signal from external disturbances. In this paper, we develop a new DNA storage architecture based on modulation encoding and decoding. Results from simulation and real data demonstrate that this storage architecture has three advantages: First, modulation encoding provides a storage friendly encoding architecture to satisfy biological constraints and have similar thermodynamic property. Second, modulation signal proves to be highly effective at detecting indels, tolerating ~40% errors. Third, the error detection and correction processes only require linear time complexity, making them extremely time efficient and thus suitable for large scale application. To the best of our knowledge, this new storage architecture far exceeds the comprehensive performance of any state-of-the-art works.

Results

Modulation-based DNA storage. Fig.1a shows the four steps of our modulation-based storage paradigm. First, the original binary information is modulated with the carrier stream to generate a DNA sequence, according to a predefined rule. Second, we transform the observed strands into marked reads by aligning them with the carrier strand to identify possible errors locations. Third, align the marked reads in each cluster and use majority voting to obtain a consensus DNA sequence. Finally, use the carrier strand to demodulate the consensus DNA sequences to recover the original binary information.

Encoding with the carrier strand. In DNA storage, the stored binary information is usually partitioned into many strands with a fixed length n. All these strands are modulated with a common carrier strand c to generate their corresponding DNA sequences. In this paper, the carrier strand c is composed of repeated substring like '1001', whose length (carrier period) is denoted as p_c. Given a binary carrier strand c='**1001**1001**1001**', a binary message strand m='010011010110' is transformed bit by bit into a DNA sequence s='**CTAC**GTAG**CTTC**', according to the following rule (shown in Fig. 1b):

(1) If $c[i]$='0' $(0 < i < n)$, it will modulate $m[i]$='0' to 'A' and $m[i]$='1' to 'T';

(2) Otherwise $c[i]$='1', it will modulate $m[i]$='0' to 'C' and $m[i]$='1' to 'G'.

Modulation encoding provides a convenient mechanism to satisfy the sequence

constraints by selecting appropriate periodic string in c. Not only can we easily strike a balance in GC content, but it can also be uniformly distributed across the sequence. For example, given c ='**100**1100**1001**', all encoded sequences will share a similar pattern of two A/T bases surrounded by a pair of G/C bases. DNA sequences with this pattern have desirable biological properties, such as similar melting temperatures and homopolymers of at most 2**.**

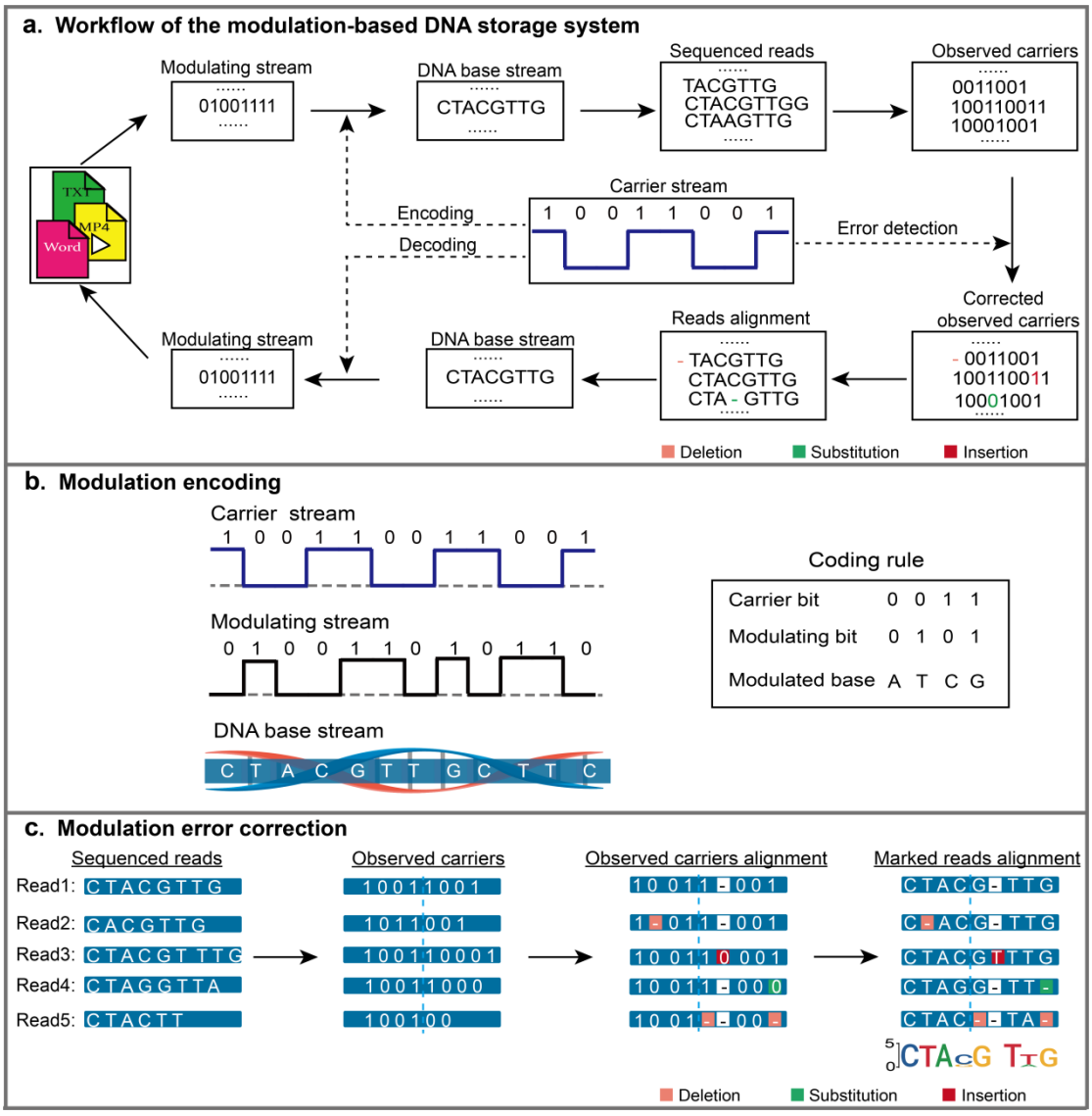

Fig. 1. Overview of the modulation-based DNA storage workflow. a Workflow of the proposed DNA storage. **b** Modulation-based encoding. **c** Modulation-based error detection and error correction.

Decoding with the carrier strand. The main challenge in decoding comes from the fact that the indels shift bases away from their original positions. Given t reads $r_1,...,r_t$ for a sequence s, the decoding problem can be formulized as inferring a consensus sequence s^* with the maximal posteriori likelihood $p(s^* \mid r_1,...,r_t)$.

According to the modulation rule, any read r_k of sequence s can be transformed

to a binary strand $c_k^{'}(1 \le k \le t)$, which we call the observed carrier strand. Obviously, $c_k^{'}$ should be similar to c in a large degree as s is produced by the carrier strand c. Based on this observation, the optimal alignment of $c_k^{'}$ to c may provide the best inference of the occurrences of IDS in read r_k. Taking c='10011001' and r_k ='CTAATAG' as an example, the best alignment of $c_k^{'}$ ='1000001' to c should be '100**0**-001' which includes one substitution in the 4th position and one deletion in the 5th position ('-' denotes deletion). This means that r_k may involve one substitution and one deletion in the corresponding positions. That is, it should be modified as $r_k^{'}$ = 'CTA**A**-TAG' and we name it as the marked read of read r_k. Then, the decoding problem $p(s^* \mid r_1,...,r_t)$ can be approximated by $p(s^* \mid r_1^{'},...,r_t^{'})$, which can be solved by a simple voting. Fig. 1c shows the main steps of the modulation-based error detection and correction (orange for deletion, red for insertion and green for substitution).

Finally, we can obtain the original message simply by demodulating the obtained consensus sequence using the opposite of the encoding rule. If $c[i]$='0', 'A' will be demodulated to $m[i]$='0' and 'T' to $m[i]$='1'; otherwise, 'C' will be demodulated to $m[i]$='0' and 'G' to $m[i]$='1'.

In sum, the carrier signal c provides a valuable prior knowledge of all encoded sequences. This information enables some powerful error detection capabilities based on the pairwise maximal likelihood alignment of a distorted $c_k^{'}$ to the template c. As we know, sequence alignment has been a thoroughly studied problem in bioinformatics: there exist famous algorithms such as the Needleman-Wunsch algorithm. In this paper, we apply a multiple sequence alignment (MSA) software MAFFT[11] to align $c_k^{'}$ and c. It should be noted that MAFFT reduces the time complexity of pairwise alignment from $O(n^2)$ to $O(n \log n)$ by fast Fourier transform. Therefore, the key process of error detection in each read can be accomplished in quasi-linear time $O(n \log n)$.

Sequence properties of modulation encoding. Fig. 2 shows how different encoding methods affect the distribution of GC content, maximum homopolymers, and melting temperatures of the DNA sequences. We compared our modulation encoding scheme with the other four classical encoding methods: Base Coding(00->A, 01->C, 10->G, 11->T), Goldman 2013[8], Church 2012[12], and DNA Fountain 2017[21]. For our method, we use the carrier strand c='1001……1001' to encode the original information. We can see that sequences by modulation have fixed GC content while others generally range between 30~70% (Fig. 2a). The maximal homopolymers in sequences by Goldman 2013[8] and modulation are 1 and 2 while other may even reach 4 (Fig. 2b). These two observations demonstrate that by selecting an appropriate carrier strand c, modulation encoding provides a convenient mechanism to satisfy the sequence constraints.

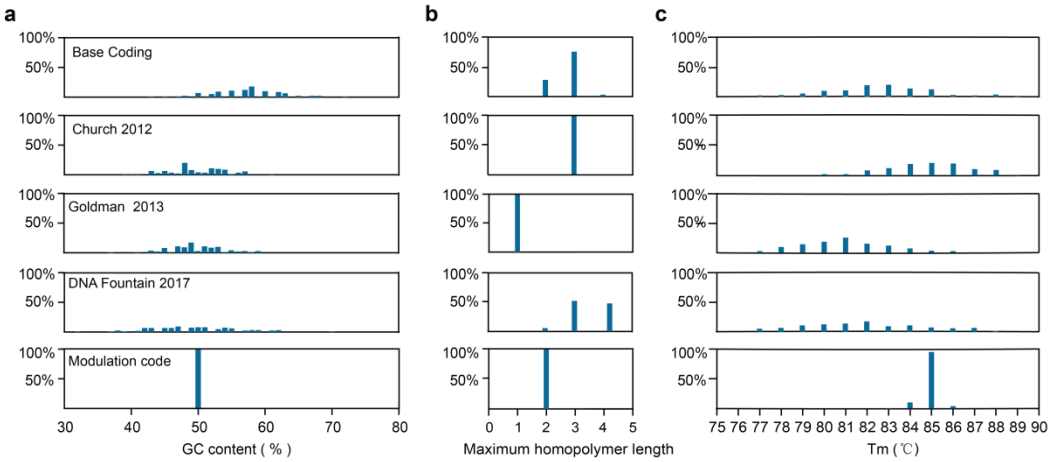

Fig. 2. Comparison of the properties of DNA sequences by Base Coding, Goldman 2013, Church 2012, DNA Fountain code 2017 and modulation-based encoding. a Distribution of GC content. **b** Distribution of maximum homopolymers. **c** Distribution of melting temperatures Tm.

The melting temperature of sequences by modulation, predicted by a web tool MFEprimer (https://mfeprimer3-0.igenetech.com/) [28], is ~85°C while others range in 77~87°C (Fig. 2c). This stable thermodynamic property is mainly attributed to the uniform distribution of GC across the sequences. Therefore, modulation provides a storage friendly encoding mode which is favorable to DNA synthesizing, PCR (Polymerase Chain Reaction), and sequencing processes. Not only can this feature reduce the occurrences of unexpected errors to some extent, but it can also improve the efficiency of the data reading processes (i.e., PCR and sequencing).

Decoding performance using different carrier periods. Intuitively, longer period carrier strands tend to include more complex patterns, and such complex patterns may help to detect abnormal indels. One way to describe the complexity of sequences is to count the number of substrings in them. For example, carrier strand c_1= '10101010' include substrings '10', '01', '101', '010', '1010', and '0101', while c_2= '10011001' includes substrings '10', '00', '01', '11', '100', '001', '011', '110', '1001', '0011', '0110', and '1100'. Obviously, the later contains more substrings than the former. Fig. 3 shows the average decoding performance by carrier strands with period 2, 4, 8, and 16 respectively.

First, the average decoding performance tends to be stable as the period increases. When the period is 2, complete recovery can only be accomplished in the lowest error rate we tested, and the average performance is below 50% in other error settings. However, the average performance improved dramatically as the period increases to 4. And it becomes stable as period increases to 8. These observations echo our assumption that longer period is beneficial to complicate error settings. However, longer periods may affect the homogeneous distribution of GC content and the control of homopolymers. Therefore, considering both the sequencing property and the error correction performance, we find that setting the period to 8 may be appropriate in most situations.

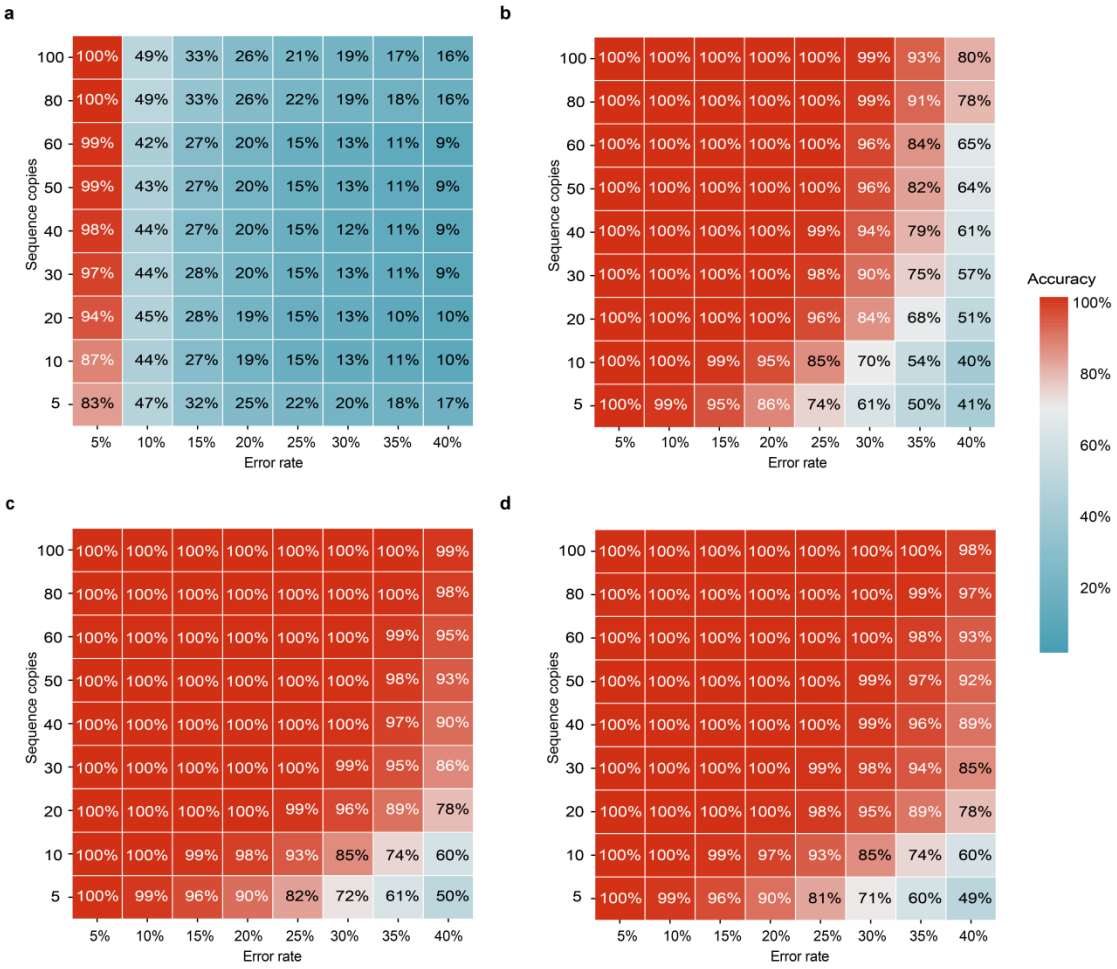

Fig. 3. Decoding performance at different carrier periods. a $p_c = 2$. **b** $p_c = 4$. **c** $p_c = 8$. **d** $p_c = 16$. The value in each colored box denotes the average recovering accuracy at the corresponding error rate and sequence copies.

Second, modulation-based decoding is extremely robust to IDSs with period 8 and 16, where given 100 sequence copies, we can even recover 99% of the data at error rate 40%. Currently, we have yet to see any reports that can deal with errors up to 40%. On the other hand, at low error rate levels, such as 5%, it can completely recover the data with just 5 sequence copies, which is far less than the minimal copies required in previous works. To the best of our knowledge, this performance far exceeds the error correction capacity of any state-of-art methods with a logical density 1bit/nt (See Table 1).

Such robustness mainly comes from the separation of error detection and error correction. In the error detection process, the pairwise alignment of c and c' can detect indels in each read at a global scale, which are more accurate and effective than those based on local information as in the works of Press et.al and Lenz et.al [23, 24]. In DNA storage, to determine where the indels occur is the key to correct errors. In the

error correction phase, the consensus sequence inference is just a majority voting at each position based on a group of reads belonging to the same sequence. As the marked reads have indicated the possible indels and some substitutions, the voting accuracy in each position is significantly improved. In sum, the combination of these two processes effectively takes advantage of the information in the carrier signal and the multiple sequences, allowing it to deal with complex IDSs in DNA storage.

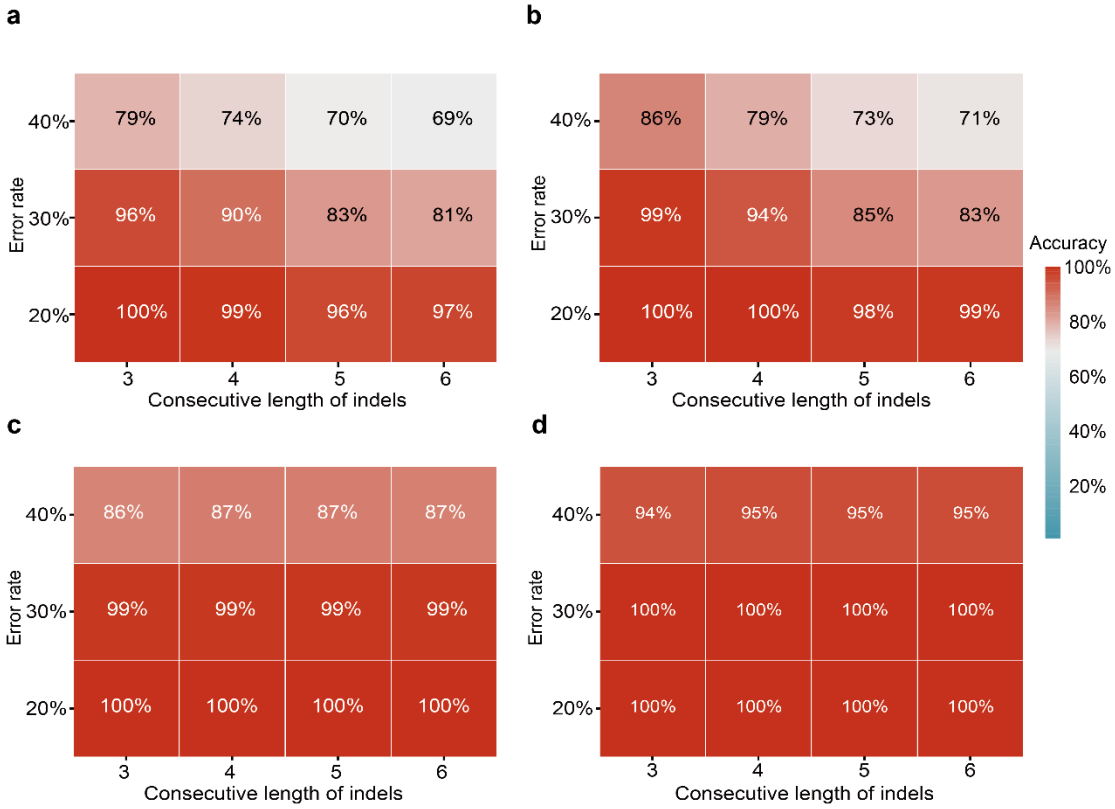

Fig. 4. Decoding performance to consecutive insertions/deletions. a period 8, sequence depth 30. **b** period 8, sequence depth 60**. c** period 16, sequence depth 30. **d** period 16, sequence depth 60.

Decoding performance for consecutive insertions/deletions. Previous studies have demonstrated that consecutive indels are frequently observed in sequenced reads, which may result in many abnormal reads with highly incorrect lengths [5, 29, 30]. Compared with a single insertion or deletion, consecutive indels are more difficult to correct. For example, HEDGES can deal with consecutive deletions, but can't tolerate consecutive insertions larger than 2 [23]. We further investigate the decoding performance of the proposed method on consecutive indels. For simplicity, we assume that insertion or deletion errors in all reads occur consecutively and have the same length.

Fig. 4 shows the average decoding performance with period 8 and 16 given sequence copies 30 and 60, respectively. As the consecutive length increases, the performance on period being 16 is better than that of period 8, especially for error rate $\geq$ 20%. As the error rate increases, longer consecutive length indels may destroy the

periodic structure of the carrier strands, which may affect its error detection capability. This is the main reason that carrier strands with period 16 are more robust than those with period 8, as the lengths of consecutive indels are far less than 16. In addition, increasing the sequence copies can improve the decoding performance at high error rates.

In most previous works, low-quality reads are usually discarded, which for large-scale application may lead to non-negligible loss both in cost and time. However, this problem can be significantly alleviated as modulation-based error detection can defend against the consecutive indels.

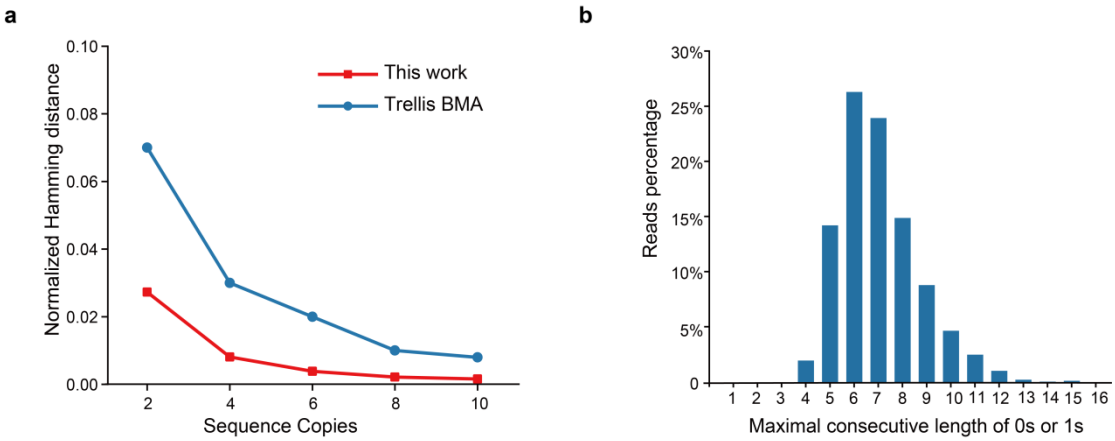

Fig. 5 Performance of the proposed method (red color) and Trellis BMA (blue color) on the real dataset. a The normalized Hamming distance at sequence copy 2,4, 6, 8, 10. **b** Distribution of the maximal consecutive 0s or 1s in the carrier strands of the encoded DNA sequences.

Decoding Performance on a Real Dataset. We compare our decoding performance with that of Trellis BMA[4] on a real dataset published by the Microsoft group. In order to apply the proposed method on this dataset, we first construct a carrier strand for each DNA sequence by translating A/T to 0 and G/C to 1. Then the decoding process is used for the reads in each cluster.

Fig. 5a shows the average normalized Hamming distance with different sequence copies. Clearly, the proposed method dramatically outperforms Trellis BMA on the real dataset even with very few copies. However, although the estimated error rate is only about 5.9%, there still exist a few uncorrected errors with sequence copies being 10. To understand this inconsistency with the results we saw in Fig. 3, we further investigate the maximal consecutive 0s or 1s in the constructed carrier strands. Fig. 5b. shows that more than 80% of the carrier strands have consecutive 0s or 1s longer than 6. That is, these encoded DNA sequences include many continuous GC or AT regions. Because of their simple patterns, such consecutive 0s or 1s may reduce the error detection capability of the carrier strands. This further verifies that simple patterns may limit the carrier strands' error detection capability.

Discussions

In this paper, we propose a modulation-based DNA storage architecture with unified encoding and decoding schemes. To take a comprehensive review, Table 1 lists the coding scheme, error correction (EC) algorithm, time complexity, reported maximal tolerated error rate, and logical density of the four previous works and ours.

Table 1. Comparison with the state-of -the-art methods

Method	Press et. al[23]	Lenz et. al[24]	Antkowiak et. al[7]	Song et. al [31]	This work
Coding Scheme	Hash + RS	CC+LDPC	RS+ RS	CRC + Anchor	No
Randomization	No	Yes	Yes	Yes	**No**
Encoding Mode	00-A 01-T		10-G 11-C		0-A/T 1-G/C
EC Algorithm	A* + RS	HMM	MSA + RS	GPS	PSA
Decoding complexity	$O(Nc^n)$	$O(Nc^n)$	$O(Nn^2)$	$O(Nc^n)$	$O(Nn\log n)$
Decoding Mode	Inner-Outer	Inner-Outer	MSA-Inner-Outer	One step	Detection-Correction
Decoding Level	binary	binary	DNA- binary	DNA	DNA
Max tolerated error	10%	18%	14.5%	10%	~40%
Logical density (bit/nt)	0.5	0.5	0.8	1.5	1
Minimum coverage	<5	20	120	100	100

Notes: 1. "RS" means Reed-Solomon code, "CC" means convolutional code, "LDPC" means low density parity check, "CRC" means cyclic redundancy check, "HMM" means hidden Markov model, "MSA" means multiple sequence alignment, "GPS" means greedy path search, and "PSA" means posterior sequence alignment. 2. N is the number of sequenced reads, n is the length of the encoded sequence, and c is a constant larger than 1.

In terms of coding scheme, our method is simple and storage friendly: it does not require randomization or adding redundancy. By selecting the appropriate carrier strand, the encoded DNA sequences not only satisfy the constraints of GC content and homopolymers, but also have similar thermodynamic properties which are beneficial to the biochemical techniques and may help to avoid the generation of errors in some degree. However, other encodings have to take hash or convolution operations[23], add RS/LDPC/CRC redundancy[7, 24] and might even perform XOR randomization on the original binary stream before translating them into DNA sequences. In addition, some encoding methods, such as the fountain code[21] and the de Brujin graph method by Song et.al [31], need a filter process to discard binary streams containing illegal subsequences .

In terms of accuracy, our method can tolerate up to 40% errors, which far exceeds the state-of-art methods. Compared to our method, the ones in Lenz et.al [24], Antkowiak et.al [7], and Press et.al [23] correct less errors (up to 18%, 14.5%, and 10%), and have lower logical densities(0.5, 0.8, and 0.5). The method by Song et.al [31] can correct up to 10% errors with a relatively high logical density of 1.5. To tolerate higher errors, these methods would have to add more logical redundancy, which will further lower their logical density. However, our method can do so by increasing the sequence copies without sacrificing logical density (See Fig. 3 C/D).

In terms of strategies for error correction, the robustness of our method roots from the effective coordination between the powerful global error detection at read level and the simple error correction in reads cluster level. For each read, not only can the error detection mechanism distinguish the uncontaminated bases, but also help to infer their most probable positions. This is the key in the subsequent multiple reads voting process, as only these conserved bases are used to infer the consensus base in each position. Even if the reads have as much as 40% random errors, only a few conserved bases are needed for each position to infer its consensus base. That is the main reason why our method has such high error tolerance. However, A* or HMM searching processes[23, 24] actually detect and correct errors utilizing the local constraints, which could make incorrect decisions. As the errors increase, the inner decoder could be overloaded, leaving more errors for the outer decoder to correct than it can handle. Although MSA, used in the work by Antkowiak et. al[7] as an inner decoder, may correct some errors including indels, previous works[9, 13] have verified that RS code, which was used as the outer decoder, can only tolerate at most 5% errors , limiting its overall capacity. The one step method by Song et. al [31] attempted to find the correct path in a de Bruijn graph which is consisted of uncontaminated DNA k-mers. However, the probability that a k-mer is uncontaminated will drop dramatically as the error rate increases. For k=18 in their work, this probability drops from 15% to 5% as error rate increases from 10% to 15%. Although increasing sequence coverage may alleviate this effect to some degree, the probability of the correct path consisted of tens or hundreds of k-mers will tend to be zero (0.05^{303}, n=320 nt). Therefore, such graph searching based method may not work for error rate larger than 15%.

In terms of time complexity, our method is the most efficient. Given N sequenced reads and their length n, it only needs N times pairwise sequence alignment (PSA) for error detection, and the following multiple sequence voting in a cluster is quasi-linear to N. It has polynomial time complexity of order $O(Nn\log n)$. The method by Antkowiak et. al[7] involves multiple sequence alignment (MSA) in each cluster and the normal RS decoding. It too has polynomial time complexity, but of order $O(Nn^2)$. The time complexities of the other three are determined by the A*, hidden Markov model (HMM), and greedy path search (GPS) algorithm, respectively. Although various heuristic strategies can be applied, they all have an exponential time complexity $O(Nc^n)$, where $c>1$ is a constant determined by the average searching branches.

At this point, the cost per bit using current DNA storage technologies is still much higher than those of traditional electronic and optical storage devices. Developing DNA storage-oriented technologies allowing more errors may provide enough room for further reducing the cost of synthesis and sequencing. Modulation-based DNA storage is characterized by storage-friendly encoding, ultra high error tolerance, and extreme efficiency in decoding. Therefore, it not only paves a solid foundation for reliable information retrieval in high error environment, but could also drive the development of low-cost synthetic technologies. We believe that this new storage architecture could facilitate the early realization of large-scale DNA storage application.

Methods

Datasets used for experiment results. For simulation experiments, a text file "The Grandmother" (excerpted from "Andersen's Fairy Tales), is encoded into 140 DNA sequences of 120 bases (8 bases for index and 112 bases for data). Error rates in the encoded sequences range between 5% ~ 40%, where insertions, deletions, and substitutions are equally likely. The sequence coverage range from 5 ~ 100. All experimental results are obtained by repeating 1,000 times under a given error rate and a fixed number of sequence copies.

The real dataset[4] includes 269,709 reads of 10,000 uniform random DNA sequences of length 110 (https://github.com/microsoft/clustered-nanopore-reads-dataset). All DNA sequences were synthesized by Twist Bioscience and sequenced using ONT MinION, and the estimated error rate in the sequenced reads is about 5.9% in total (p_{ins}=1.7%, p_{del}=2%, p_{sub}=2.2%). The noisy reads were grouped by a pseudo clustering algorithm[32].

Construction of the periodic carrier strands. In this paper, we investigate the performance of carrier strands with period $p_c = 2, 4, 8, 16$. To satisfy the constraints of GC content and maximal homopolymers, the period substrings in the carrier strands should satisfy the following two criterions:

(1) The percentage of 1s (or 0s) should be 50%.

(2) The consecutive length of 1s (or 0s) should be less than 4.

For period 2, there are only two carrier strands: '0101…0101' and '1010… 1010', which constitute of substrings '01' and '10'. For period 4, there are 4 carrier strands : '0110', '1001', '1100', and '0011', . For period 8, we enumerate all binary strings with length 8, and discard those with period 2 and 4. Substrings for period 16 is obtained in the same way.

Error-correction for sequenced reads. The proposed error-correction process is illustrated in Fig. 1c, and it contains the following steps: Step 1, for each reads cluster, derive the observed carrier strand of the reads according to the modulation rule. Step 2, obtain the marked read by using MAFFT to align the observed carrier strand to the carrier strand[11]. Step 3, deduce the consensus sequence for each cluster of marked reads using a simple voting strategy. In the voting process, bases that are marked as insertion, deletion, or substitution errors should not be considered. Finally, the consensus sequences can be demodulated into the binary data by reversing the encoding rule.

Data availability

All data are available in the main text or the supplementary materials.

Code availability

Code can be downloaded from
https://github.com/BertZan/Modulation-based-DNA-storage

References

1. Xu C, Ma B, Gao Z, Dong X, Zhao C, Liu H. Electrochemical DNA synthesis and sequencing on a single electrode with scalability for integrated data storage. *Science Advances* **7**, eabk0100 (2021).

2. Qian L, Ouyang Q, Ping Z, Sun F, Dong Y. DNA storage: research landscape and future prospects. *National Science Review* **7**, 1092-1107 (2020).

3. Zan X*, et al.* A Hierarchical Error Correction Strategy for Text DNA Storage. *Interdisciplinary Sciences: Computational Life Sciences* **14**, 141-150 (2022).

4. Srinivasavaradhan SR, Gopi S, Pfister H, Yekhanin S. Trellis BMA: coded trace reconstruction on IDS channels for DNA storage. (2021).

5. Cretu Stancu M*, et al.* Mapping and phasing of structural variation in patient genomes using nanopore sequencing. *Nature Communications* **8**, 1326 (2017).

6. Organick L*, et al.* Random access in large-scale DNA data storage. *Nat Biotechnol* **36**, 242-248 (2018).

7. Antkowiak PL*, et al.* Low cost DNA data storage using photolithographic synthesis and advanced information reconstruction and error correction. *Nature Communications* **11**, 5345 (2020).

8. Goldman N*, et al.* Towards practical, high-capacity, low-maintenance information storage in synthesized DNA. *Nature* **494**, 77-80 (2013).

9. Meiser LC*, et al.* Reading and writing digital data in DNA. *Nat Protoc* **15**, 86-101 (2019).

10. Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res* **32**, 1792-1797 (2004).

11. Katoh K, Standley DM. MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution* **30**, 772-780 (2013).

12. Church GM, Gao Y, Kosuri S. Next-Generation Digital Information Storage in DNA. *Science* **337**, 1628 (2012).

13. Grass RN, Heckel R, Puddu M, Paunescu D, Stark WJ. Robust Chemical Preservation of Digital Information on DNA in Silica with Error‐Correcting Codes. *Angew Chem Int Ed Engl* **54**, 2552-2555 (2015).

14. Chen W*, et al.* An artificial chromosome for data storage. *National Science Review*, (2021).

15. Blawat M*, et al.* Forward Error Correction for DNA Data Storage. *Procedia Computer Science* **80**, 1011-1022 (2016).

16. Chen WG, Wang LX, Han MZ, Han CC, Li BZ. Sequencing barcode construction and identification methods based on block error-correction codes. *Sci China Life Sci* **63**, 1580-1592 (2020).

17. Xue TB, Lau FCM. Construction of GC-Balanced DNA With Deletion/Insertion/Mutation Error Correction for DNA Storage System. *Ieee Access* **8**, 140972-140980 (2020).

18. Takahashi CN, Nguyen BH, Strauss K, Ceze L. Demonstration of End-to-End Automation of DNA Data Storage. *Scientific Reports* **9**, 4998 (2019).

19. Deng L*, et al.* Optimized Code Design for Constrained DNA Data Storage With Asymmetric Errors. *Ieee Access* **7**, 84107-84121 (2019).

20. Lu XZ*, et al.* Error Rate-Based Log-Likelihood Ratio Processing for Low-Density Parity-Check Codes in DNA Storage. *Ieee Access* **8**, 162892-162902 (2020).

21. Erlich Y, Zielinski D. DNA Fountain enables a robust and efficient storage architecture. *Science* **355**, págs. 950-954 (2017).

22. Bornholt J, Lopez R, Carmean DM, Ceze L, Seelig G, Strauss K. Toward a DNA-Based Archival Storage System. *IEEE Micro* **37**, 98-104 (2017).

23. Press WH, Hawkins JA, Jones SK, Schaub JM, Finkelstein IJ. HEDGES error-correcting code for DNA storage corrects indels and allows sequence constraints. *P Natl Acad Sci USA* **117**, 18489-18496 (2020).

24. Lenz A, Maarouf I, Welter L, Wachter-Zeh A, Amat A. Concatenated Codes for Recovery From Multiple Reads of DNA Sequences. (2020).

25. Zhan YJ, Huang FC. Generalized Spatial Modulation With Multi-Index Modulation. *Ieee Commun Lett* **24**, 585-588 (2020).

26. Moore BC, Sek A. Effects of carrier frequency, modulation rate, and modulation waveform on the detection of modulation and the discrimination of modulation type (amplitude modulation versus frequency modulation). *The Journal of the Acoustical Society of America* **97**, 2468-2478 (1995).

27. Agrawal. Modulation instability induced by cross-phase modulation. *Physical review letters* **59**, 880-883 (1987).

28. Wang K*, et al.* MFEprimer-3.0: quality control for PCR primers. *Nucleic Acids Res* **47**, W610-W613 (2019).

29. van Dijk EL, Jaszczyszyn Y, Naquin D, Thermes C. The Third Revolution in Sequencing Technology. *Trends Genet* **34**, 666-681 (2018).

30. Jain M*, et al.* MinION Analysis and Reference Consortium: Phase 2 data release and analysis of R9.0 chemistry. *F1000Res* **6**, 760-760 (2017).

31. Song L, Geng F, Gong Z, Li B, Yuan Y. Super-robust data storage in DNA by de Bruijn graph-based decoding. *bioRxiv*, 2020.2012.2020.423642 (2020).

32. Rashtchian，*, et al.* Clustering Billions of Reads for DNA Data Storage. *Advances in Neural Information Processing Systems* **30**, (2017).

Acknowledgements

This work was supported by the National Natural Science Foundation of China (grant nos.62072128 and 62002079).

Author contributions

W.B.L. and P.X. supervised the research. W.B.L. and X.Z.Z. conceived the concept. W.B.L. managed coauthor contributions to the paper. X.Z.Z. wrote the Python codes, performed the simulations and analyzed the data. R.Z.X. polished the paper. R.Z.X. and X.Y.Y. discussed on the data. All authors contributed to the writing of the paper.

Competing interests

The authors declare no competing interests.

Competing interests

The authors declare no competing interests.